Hindawi
Security and Communication Networks
Volume 2021, Article ID 5263726, 10 pages
https://doi.org/10.1155/2021/5263726

Research Article

Deep Grid Scheduler for 5G NB-IoT Uplink Transmission

Han Zhong,[1,2] **Ruize Sun**,[1] **Fengcheng Mei**,[1] **Yong Chen**,[1] **Fan Jin**,[3] **and Lei Ning**[1]

[1]*College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China*
[2]*College of Applied Technology, Shenzhen University, Shenzhen, China*
[3]*Shenzhen Winoble Technology Co., Ltd, Shenzhen, China*

Correspondence should be addressed to Lei Ning; ninglei@sztu.edu.cn

Received 18 July 2021; Accepted 5 August 2021; Published 15 August 2021

Academic Editor: Xin Liu

Since the birth of narrowband Internet of Things (NB-IoT), the Internet of Things (IoT) industry has made a considerable progress in the application for smart cities, smart manufacturing, and healthcare. Therefore, the number of UEs is increasing exponentially, which brings considerable pressure to the efficient resource allocation for the bandwidth and power constrained NB-IoT networks. In view of the conventional algorithms that cannot dynamically adjust resource allocation, resulting in a low resource utilization and prone to resource fragmentation, this paper proposes a double deep Q-network (DDQN)-based NB-IoT dynamic resource allocation algorithm. It first builds an NB-IoT environment model based on the real environment. Then, the DDQN algorithm interacts with the NB-IoT environment model to learn and optimize resource allocation strategies until it converges to the optimum. Finally, the simulation results show that the DDQN-based NB-IoT dynamic resource allocation algorithm is better than the traditional algorithm in the resource utilization, average transmission rate, and UE average queuing time.

1. Introduction

With the advancement of science and technology, the IoT is being used more and more widely in various industries [1]. In order to meet the needs of the IoT industry, it has redesigned a new communication solution for the IoT, which is called the NB-IoT [2]. The access system was proposed in the 69th plenary meeting of the 3rd generation (3GPP) organization. The NB-IoT system focuses on low-complexity and low-throughput radio access technology. The main research goals include improved indoor coverage, support for a large number of low-throughput user equipment, lower latency sensitivity, and ultralow equipment cost, low equipment power consumption, and network architecture. For the use scenarios of NB-IoT, the relevant technical characteristics of NB-IoT are as follows [3]:

(1) Low power consumption: NB-IoT uses power save mode (PSM) and extended discontinuous reception (eDRX) to reduce power consumption. It is estimated that a 5Wh battery can provide maximum battery life up to 10 years.

(2) Channel bandwidth: the bandwidth is 200 kHz, including a guard band of 20 kHz.

(3) Coverage enhancement: NB-IoT achieves coverage enhancement mainly by increasing the uplink power spectral density and repeated transmission so that outdoor coverage is large and indoor penetration is improved. In the same frequency band, compared with the existing general packet radio service, general packet radio service (GPRS) network coverage increased by 20 dB, the maximum coupling loss (MCL) can reach 164 dB, and the coverage area is expanded 100 times.

(4) A large number of equipment access: it can support a large number of low-throughput terminals, up to 50 K connections per cell. Under the coverage of the same base station, NB-IoT can support up to 50–100 times the number of access devices compared to the existing wireless technology.

(5) Low cost: NB-IoT only supports FDD half-duplex mode, which is cheaper than full duplex. The cost of the module is less than US$5. It is expected to be

reduced to US$2-3 by 2020, and a single antenna is used for transmission. It can also reduce the complexity of chip processing, thereby reducing costs.

Based on the characteristics above, NB-IoT can locate devices with poor channel transmission conditions or delay tolerance and can be widely used in smart homes, smart cities, smart grids, healthcare, smart manufacturing, and smart logistics [4–6].

Although NB-IoT technology is developing in full swing, it still faces the challenges of spectrum efficiency, system capacity, and interference coexistence [7]. NB-IoT introduces processes such as repeated transmission and reinitialization of the scrambling sequence, which increases the complexity of part of the NB-IoT physical layer processing process [8]. Narrowband physical uplink shared channel (NPUSCH) has a bandwidth of only 180 kHz. It not only carries uplink data services but also needs to transmit response information indicating whether the narrowband physical downlink shared channel (NPDSCH) has been successfully received. Therefore, efficient use of spectrum resources is very necessary [9].

Therefore, how NB-IoT allocates spectrum resources for UEs efficiently is a key issue. In view of the low resource utilization of traditional resource allocation algorithms, fragmentation is prone to occur, and the average waiting delay is high, and this article proposes a DDQN-based dynamic resource allocation algorithm. This article first builds an NB-IoT environment model based on the real environment. The DDQN algorithm interacts with the NB-IoT environment model and stores historical experience in the experience pool to accumulate data for subsequent model training and learning. After learning and iteration, the algorithm is better than the traditional algorithm in resource utilization, average transmission speed, and average waiting time.

The rest of the paper is organized as follows. A relate work on NB-IoT is provided in Section 2. Section 3 presents the experimental design, including the environment model of NB-IoT uplink and the proposed of DDQN algorithm. The performance analysis is reported in Section 4. We finally conclude the paper in Section 5.

2. Relate Work

At present, most machine-type communications are still based on the LTE scheduler [10]. Using the LTE scheduler can maximize the overall transmission success rate and minimize the machine-to-machine (M2M) delay. In [11], EDDF-based LTE scheduling procedures have been shown to be effective in maximizing the transmission success rate and minimizing delay. In [12], Afrin et al. proposed a schedular based on an OPNET simulation model of an LTE TDD system. The scheduler can satisfy the uplink delay budget for more than 99% of packets for bursty delay sensitive M2M traffic even when the system is fully loaded with regard to the data channel utilization. However, this requires additional signaling overhead to transmit the waiting time at the head of the queue.

In order to solve the static problem of the algorithms, adaptive algorithms have also been extensively studied in the resource allocation of NB-IoT. In [13], Sampath et al. used an adaptive algorithm to develop an analytical outer loop power control model to deal with signal-to-interference ratio fluctuations. Li et al. [14] studied the influence of uplink interference on link adaptation in heterogeneous networks and proposed a cooperative uplink adaptation scheme using cooperation between base stations. In [15], a novel algorithm for improving outer loop link adaptation (OLLA) convergence speed in the downlink of long-term evolution (LTE) is presented. The algorithm is validated with a connection-level simulator, fed with real connection traces collected from a live LTE network. In [16], the potential of OLLA to cope with the aforementioned problem is studied, and a dynamic OLLA (d-OLLA) algorithm is proposed.

In order to optimize the resource allocation of NB-IoT, some resource issues for NB-IoT have also been studied. Su et al. [17] propose a method for active detection and processing of redundant rules. In [18], a new Aloha-based tag identification protocol is presented to improve the reading efficiency of the EPC C1 Gen2-based UHF RFID system. Min Oh et al. [19] proposed an efficient small data transmission scheme in the 3GPP NB-IoT system. For the efficient use of radio resources, the proposed scheme enables devices in an idle state to transmit a small data packet without the radio resource control connection setup process. This can improve the maximum number of supportable devices in the NB-IoT system which has insufficient radio resources. Recently, Huang et al. [20] identified radio resource scheduling issues for NB-IoT systems and provided a comprehensive performance evaluation. Then, the authors proposed an NB-IoT downlink scheduling algorithm. Wu et al. [21] proposed a deep Q-learning network (DQN) method used to control the hand-over (HO) procedure of the user equipment (UE) by well capturing the characteristics of wireless signals/interferences and network load. In [22], a multiagent deep Q-network- (DQN-)based dynamic joint spectrum access and mode selection (SAMS) scheme is proposed for the SUs in the partially observable environment. Zhang et al. [23] proposed a two-step deep reinforcement learning-based algorithm to solve nonconvex and dynamic optimization problem. However, the above research studies have not addressed how to reduce resource fragmentation. In our previous work [24], we propose a dynamic resource allocation algorithm without theoretical analyses for the NB-IoT uplink scheduling problem and did not consider the waiting delay of the UE.

In this article, we consider the waiting delay of the UE and the NB-IoT scheduling problem for 3GPP NB-IoT cellular networks. The objective is to maximize resource utilization and reduce resource fragmentation while ensuring UE has a short waiting time delay. Therefore, we propose a dynamic resource scheduling algorithm based on deep reinforcement learning to optimize the resource utilization of NB-IoT. In Section 3, we will first introduce the system model and the problem formulation will be presented as follows.

3. System Model and Problem Formulation

In this section, we first introduce the NB-IoT uplink system model and give some parameter settings in the NB-IoT uplink. Then, we will model and analyze the resource allocation problem based on the NB-IoT uplink system model.

3.1. The System Model of NB-IoT Uplink.

NB-IoT uplink frequency domain resources are the same as downlink, and frequency domain resources are 180 kHz, using SC-FDMA [3]. Taking into account the low-cost requirements of NB-IoT devices, it is necessary to support single frequency (single tone) transmission in the uplink. In addition to the original 15 kHz, a new subcarrier spacing of 3.75 kHz has been set for a total of 48 subcarriers. Considering that 3.75 kHz is rarely used in real commercial environments, this article only considers the case of 15 kHz, which is divided into 12 subcarriers in the frequency domain. For the uplink, NB-IoT defines two physical channels: NPUSCH and NPRACH (narrowband physical random access channel) and demodulation reference signal (DMRS). NPRACH allocation in the frequency domain is periodically allocated by the evolved node B (eNB). The UE transmits the NPRACH on the fixed frequency domain resources allocated by the eNB, and the remaining channels are used for NPUSCH transmission. NPUSCH is used to transmit uplink data and control information. NPUSCH transmission can use single tone or multitone transmission.

Compared with the physical resource block (PRB) as the basic resource scheduling unit in long-term evolution (LTE), the resource unit of the NB-IoT uplink shared physical channel NPUSCH is scheduled with a flexible combination of time-frequency resources. The basic unit of scheduling is called resource unit (RU). NPUSCH has two transmission formats, the corresponding resource units are different, and the content of transmission is also different. NPUSCH format 1 is used to carry the uplink-shared transmission channel UL-SCH, to transmit user data or signaling, and the UL-SCH transmission block can be scheduled and sent through one or several physical resource units. The occupied resource unit includes two formats, which are single tone and multitone. NPUSCH format 2 is used to carry uplink control information, such as ACK/NAK response. The specific RU of single tone and multitone is defined in Table 1.

The value of RU in NPUSCH is determined by TBS, MCS, and the number of repetitions. The specific RU value is calculated from Table 2.

3.2. Problem Formulation.

In this section, we study the NB-IoT uplink resource allocation problem over NB-IoT networks. The objective is to maximize the resource utilization of NB-IoT and reduce resource fragmentation. The system model can be formulated as follows.

First of all, NB-IoT does not support measurement reports. According to the difference in minimum path loss (MCL), 3GPP defines three coverage levels: normal coverage, extended coverage, and extreme coverage. The MCL corresponding to the three coverage levels is no higher than 144 dB, no higher than 154 dB, and no higher than 164 dB. Under different coverage levels, NPUSCH and NPRACH channels use different MCS and repetition times.

NPRACH is periodically transmitted in the NB-IoT uplink, and the number of repetitions $N_{\text{rep}}^{\text{NPRACH}}$ corresponding to the three coverage levels is 2, 4, and 8. The unit length of NPRACH $N_{\text{slots}}^{\text{NPRACH}}$ is 2 ms. And, the number of PRACH subcarriers $N_{\text{sc}}^{\text{NPRACH}}$ is configured by the base station; then, the resource R_{NPRACH} occupied by a NPRACH resource can be expressed as

$$R_{\text{NPRACH}} = N_{\text{rep}}^{\text{NPRACH}} * N_{sc}^{\text{NPRACH}} * N_{\text{slots}}^{\text{NPRACH}}. \quad (1)$$

The starting subcarrier index of the NPRACH resource $N_{\text{scoff set}}^{\text{NPRACH}}$ is configured by the base station, and the default base station configuration is used in this article, which is 8, 16, and 32.

The UE transmits data through the NPUSCH, and the relevant parameters of the NPUSCH are indicated by the Format N0 of the DCI in the NPDCCH, including the modulation and coding scheme I_{TBS}, the number of repetitions $N_{\text{rep}}^{\text{NPUSCH}}$, the number of time slots $N_{\text{slot}}^{\text{UL}}$, and the subcarrier indication $N_{\text{sc}}^{\text{NPUSCH}}$. From the modulation and coding scheme I_{TBS} and the data size DS to be transmitted, we can check Table 2 and calculate N_{R}:

$$N_{\text{RU}} = \arg\min\left[\text{TBS}\left(I_{\text{TBS}}\right) > \ = \text{DS}\right]. \quad (2)$$

Then, the resource R_{NPUSCH} occupied by a NPUSCH transmission can be calculated by the formula

$$R_{\text{NPUSCH}} = N_{\text{RU}}^{*} N_{sc}^{\text{NPUSCH}} * N_{\text{slot}}^{\text{UL}} * N_{\text{rep}}^{\text{NPUSCH}}. \quad (3)$$

The utilization of frequency domain resources on a time domain resource U_i can be expressed as the sum of NPRACH and NPUSCH occupying the time domain resource at that moment divided by the number of NB-IoT subcarriers. The formula is as follows:

$$U_i = \frac{\sum R_i^{\text{NPUSCH}} + \sum R_i^{\text{NPRACH}}}{N_{sc}^{RA}}. \quad (4)$$

Constrained by the communication protocol [25], the allocation of wireless resources needs to consider the period, so the goal we pursue is to maximize U_i at each moment and minimize the fragmentation of resources at each moment:

$$\left(a_{sc}, a_t\right) = \arg\max U_i. \quad (5)$$

4. Dynamic Resource Allocation Algorithm Based on DDQN

Reinforcement learning is one of the important tools in the field of machine learning. It is widely used to deal with Markov dynamic programming problems [26, 27]. As shown in Figure 1, the AI engine is designed as an agent that combines deep learning and reinforcement learning. The agent interacts with the Mac layer in NB-IoT and observes the resource occupancy and UE request as the state of the environment from the Mac layer. The AI engine generates

Table 1: NPUSCH RU format.

NPUSCH Format	Subspacing (kHz)	Sub Num	TS Num	Duration (ms)
1	3.75	1	16	32
1	15	1	16	8
1	15	3	8	4
1	15	6	4	2
1	15	12	2	1
2	3.75	1	4	8
2	15	1	4	2

Table 2: Transport block size (TBS) table for NPUSCH.

				N_{RU}				
I_{TBS}	1	2	3	4	5	6	7	8
0	16	32	56	88	120	152	208	256
1	16	32	56	88	120	152	208	256
2	32	72	144	176	208	256	328	424
3	32	72	144	176	208	256	328	424
4	56	120	208	256	328	408	552	680
5	56	120	208	256	328	408	552	680
6	88	176	256	392	504	600	808	1000
7	104	224	328	472	584	712	1000	1224
8	120	256	392	536	680	808	1096	1384
9	136	296	456	616	776	936	1256	1544
10	144	328	504	680	872	1000	1384	1736

corresponding actions to allocate corresponding resources to the UE.

The agent optimizes and adjusts the strategy through the rewards of environmental feedback and repeats this process until the optimal strategy is finally obtained. In reinforcement learning, Q learning is a very effective learning method and is widely used in various fields. Different from the SARSA algorithm and other on-policy algorithms, Q learning is updated according to the improved strategy of $q(s_{t+1}, *)$, so as to achieve closer target value. Its goal formula can be defined as

$$U_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \max_{a\in A\left(S_{t+n}\right)} q\left(S_{t+n}, a_{(sc,t)}\right). \tag{6}$$

Q learning updates the action value based on the above formula, which easily leads to maximization deviation and makes the estimated $q^{(0)}(\cdot,\cdot)$ action value too large. Therefore, double Q learning is introduced. The double Q learning algorithm uses two independent action value estimates and $q^{(1)}(\cdot,\cdot)$ and replaces $\max_a q(S_{t+1}, a_{(sc,t)})$ in Q learning with $q^{(0)}(S_{t+1}, \arg\max_a q^{(1)}(S_{t+1}, a_{(sc,t)}))$ or $q^{(0)}(S_{t+1}, \arg\max_a q^{(1)}(S_{t+1}, a_{(sc,t)}))$. Since $q^{(0)}$and $q^{(1)}$ are independent estimates, there is

$$E\left[q^{(0)}\left(S_{t+1}, A^*\right)\right] = q\left(S_{t+1}, \arg\max_a q^{(1)}\left(S_{t+1}, a_{(sc,t)}\right)\right). \tag{7}$$

In the process of double learning, both $q^{(0)}$ and $q^{(1)}$ are updated gradually, and each step of learning can select any one of the following two to update with equal probability:

$$\begin{aligned} U_t^{(0)} &= R_{t+1} + \gamma q^{(1)}\left(S_{t+1}, \arg\max_a q^{(0)}\left(S_{t+1}, a_{(sc,t)}\right)\right),\\ U_t^{(1)} &= R_{t+1} + \gamma q^{(0)}\left(S_{t+1}, \arg\max_a q^{(1)}\left(S_{t+1}, a_{(sc,t)}\right)\right). \end{aligned} \tag{8}$$

The traditional Q learning algorithm can effectively obtain the optimal strategy when the state space and action space are small. However, in actual situations, the state space and action space of the agent are very large. At this time, it is difficult for the Q learning algorithm to achieve the ideal effect. Therefore, a deep Q network composed of a combination of Q learning and neural network can solve this problem well.

Reinforcement learning data is usually nonstatic, nonindependent, and uniformly distributed. One state of data may continue to flow in, and the next state is usually highly correlated with the previous state. Therefore, small deviations in the value of the Q function will affect the entire strategy. As a supervised learning model, deep neural networks require data to meet independent and identical distribution. In order to break the correlation between data, DQN adopts the method of experience replay, storing the past training data in the form of (s_t, a_t, r_t, s_{t+1}) in the experience pool, and randomly extracts part of the data each time as the input of the neural network for training. Through the use of experience replay, the correlation between the original data is broken, and the training data become more independent and evenly distributed.

The network composed of double Q learning and DQN is called double deep Q network (DDQN). In the DDQN, only the evaluation network is used to determine the action, and the target network is used to determine the estimate of the return. The algorithm process of the DDQN is shown in Algorithm 1.

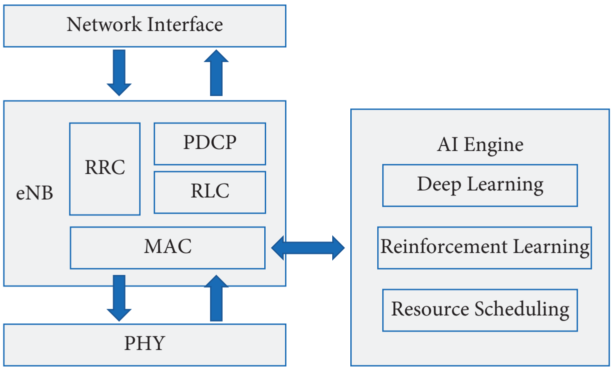

Figure 1: The interaction process between AI engine and NB-IoT environment.

```
 Initialize the evaluation network $q(\cdot,\cdot;w)$ and target network $q(\cdot,\cdot;w_{\text{target}})$
for episode in episodes do
   initialize and choose state $S_{(\text{NPUSCH, NPRACH, UE})}$ from NB-IoT MAC
   while episode not end do
      sampling $q(S,\cdot;w)$ and get action $A_{\text{sc},t}$
      observation from NB-IoT MAC and get reward $R$ and next state $S'$
      store experience $(S, A_{\text{sc},t}, R, S') \longrightarrow D$
      $D \longrightarrow (S_i, A_{(\text{sc},t)i}, R_i, S_i')_{(i\in B)}$
      $U_i = R_i \cdot + \gamma q(S_i', \arg\max_a q(S_i', a_{\text{sc},t}; w); w_{\text{target}})$
      update $w$
      $S \leftarrow S'$
      if batch size $\geq$ memory capacity then
         update $w_{\text{target}} \leftarrow w$
      end if
   end while
end for
```

Algorithm 1: Procedure of double deep Q network.

In view of the characteristics of NB-IoT uplink data transmission, the state observed by the agent is divided into 12 frequency domains, and the distribution of resource utilization in each frequency encounter on the time axis is used as the feature of each frequency domain. At the same time, the state also includes UE data size, NPUSCH format, transmission quality, number of repetitions, and the number of N_{RU}. The corresponding action taken by the agent is to allocate the corresponding resources required by the UE in the frequency domain resources. Therefore, the action of the agent is composed of 12 actions, corresponding to the divided 12 frequency domain positions.

5. Performance Evaluation

In this section, we present the analysis of NB-IoT dynamic resource allocation algorithm based on DDQN. First, we introduce the NB-IoT environment model parameters and the DDQN algorithm model parameters, and then, we will give a specific performance comparison between the DDQN algorithm and the traditional algorithm.

5.1. Model Parameter Settings. In order to verify whether the DDQN algorithm can achieve better results in a complex network environment, this paper simulates the NB-IoT uplink data link NPUSCH to establish an environment model. Every second, an average of 1000 UEs need to establish a connection to transmit data, and the communication quality of each UE is randomly selected. According to the distribution of communication quality in real scenarios [5], this article divides the communication quality into three types: good, medium, and poor, and their probability corresponds to 60%, 30%, and 10%. When each UE establishes a connection, the communication quality is randomly determined according to the probability. According to the characteristics of NB-IoT data transmission in real life, the data transmission volume is generally 50-250 bytes, and the data volume of the UE in the simulation is also randomly generated based on this range. The simulation model and deep reinforcement learning constructed in this paper are implemented by Python, and the DDQN algorithm is designed and trained based on PyTorch. The values of the network parameters and deep reinforcement learning parameters in this experiment are shown in Tables 3 and 4. The neural network used for training is a fully connected neural network, which contains a hidden layer, and the hidden layer contains 50 neurons. The activation function used by each neuron is ReLU. The size of the discount factor determines

Table 3: NPUSCH parameters.

Parameter name	Parameter value
Channel bandwidth	180
Subcarrier spacing	15
I_{TBS}	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
NPUSCH repetitions	0, 8
Data size	Range (50, 250)

Table 4: DDQN algorithm parameters.

Parameter name	Parameter value
Greedy policy, ε	0.9
Reward discount, γ	0.9
Learning rate, α	0.01
Target update frequency	100
Batch size	256
Memory capacity	5000
Actions	12
States	13

how much the algorithm attaches importance to current returns and future returns. The smaller the discount factor is, the more the algorithm tends to have short-term high returns. Since a series of actions need to be made in this experiment, in order to obtain a longer-term high return, this article sets the discount factor to 0.9.

Based on the parameters of Tables 3 and 4, the experiment scenario is simulated. In this experiment, dynamic resource scheduling is performed on the 180 kHz NB-IoT uplink data link NPUSCH. On average, 1,000 UEs are scheduled per second, and the number of iteration rounds is 100,000. Calculate the average resource utilization, average transmission speed, and average waiting delay after each dynamic scheduling as the performance evaluation index. Specific performance comparison will be shown in the following.

5.2. Analysis of NB-IoT Dynamic Allocation Algorithm Based on DDQN. In this simulation, 1,000 UE request data transmission every second, and the communication quality distribution of UE is designed to be 60% with good quality, 30% with medium communication quality, and 10% with poor communication quality. In 100,000 iterations, the actual UE distribution per second is shown in Figure 2. The figure shows the average distribution of three communication qualities per second within 100s. The three communication qualities are all at 60% and 30% and 10% fluctuate, of which good communication quality is slightly higher than 60%.

According to the data size sent by NB-IoT, in reality, the amount of data transmitted by the UE is set to be randomly distributed between 50 and 250. In this simulation, the average data size actually transmitted per second is shown in Figure 3. Compared with the average data size of good communication quality and medium communication quality, the average transmission data size of UEs with poor communication quality fluctuates greatly. The three types of data size variance are shown in Figure 4. This phenomenon is because the number of UEs is small, and the randomly distributed number is much lower than the number with good communication quality and medium communication quality.

In this simulation, the UE will use the communication quality and data volume shown in Figures 2 and 4 as the UE's communication characteristics and, respectively, use DDQN dynamic resource allocation algorithm and traditional resource allocation algorithm for resource allocation. Calculate the average resource utilization rate, average transmission speed, and average waiting time delay after each dynamic scheduling as evaluation indicators. The simulation results are shown in Figures 5–7.

As shown in Figure 5, it is a comparison chart of the average resource utilization between the DDQN dynamic resource allocation algorithm and the traditional algorithm. The resource utilization rate is obtained by calculating the utilization of the 12 subcarriers divided by the 180 kHz frequency domain. The average resource utilization rate is obtained by dividing the resource utilization rate per millisecond by the total number of UEs in history.

Through comparison, it can be found that, in the initial 10,000 iterations, the resource utilization rate of the DDQN dynamic resource allocation algorithm fluctuates greatly, and the agent is still in the exploratory stage, and the average resource utilization rate fluctuates sharply between 50% and 80%. Between 10,000 iterations and 20,000 iterations, the fluctuation of the DDQN dynamic resource allocation algorithm has been greatly reduced, and its resource utilization rate is better than that of the transmission resource allocation algorithm. After 20,000 iterations, the average resource utilization of the DDQN dynamic resource allocation algorithm can be stabilized at about 83%, which is an improvement of about 7% compared to the traditional dynamic resource allocation algorithm.

Figures 6 and 7 show the corresponding data transmission speed and the average waiting time of the UE. The sum of the data size carried by each subcarrier in one of the time domains is used as the data transmission speed per

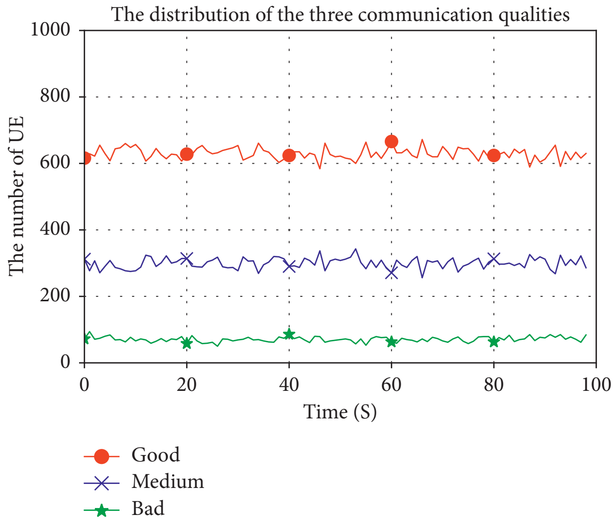

Figure 2: The distribution of the three communication qualities.

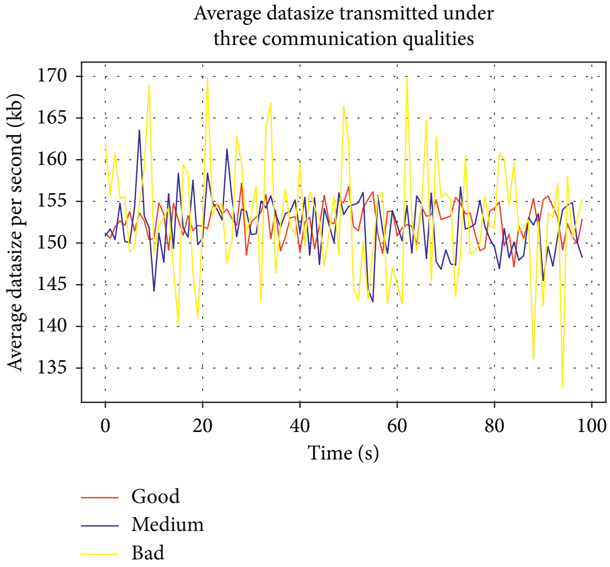

Figure 3: Average data size transmitted under three communication qualities.

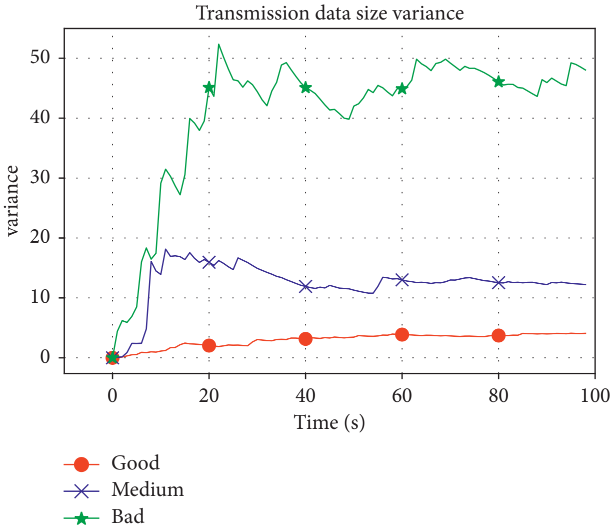

Figure 4: Transmission data size variance.

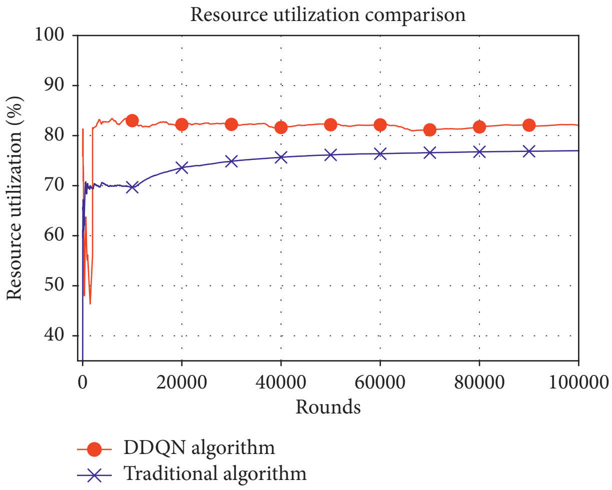

Figure 5: Resource utilization comparison.

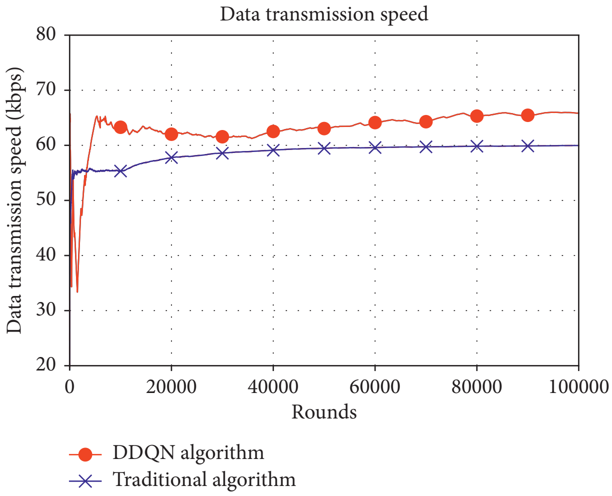

Figure 6: Data transmission speed.

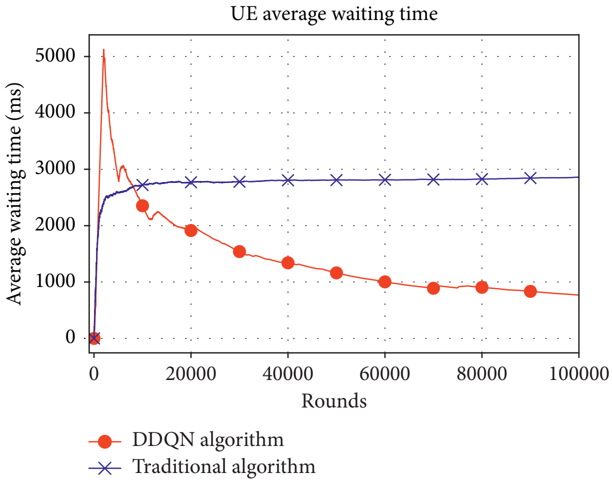

Figure 7: UE average waiting time.

millisecond, and the sum of historical data is calculated and the average value is taken as the data transmission speed at each moment. The period from the UE sending the request to the beginning of data transmission is defined as the waiting time of the UE, and the average value of the sum of the historical waiting time of the UE is taken as the average waiting time of the UE at each moment.

When the number of iterations is before 10,000, the resource utilization rate fluctuates drastically, the corresponding data transmission speed also fluctuates drastically, and the average waiting time of the UE is longer. When the number of iterations is between 10,000 and 20,000, the data transmission speed increases exponentially due to the increase in resource utilization and finally stabilizes at about 65, which is a 5% increase compared to traditional resource allocation algorithms. After the DDQN resource allocation algorithm is stable after 60,000 iterations, the average waiting time of the UE is less than 1, which is a 66% improvement compared to the traditional resource allocation algorithm.

In this simulation, we compared the DDQN dynamic resource allocation algorithm and the traditional resource allocation algorithm from the three dimensions of average resource utilization, average transmission speed, and average waiting time. In the simulation results, the DDQN algorithm is better than the traditional resource allocation algorithm in three aspects. DDQN improves the resource utilization of NB-IoT, reduces resource fragmentation, and increases the transmission speed and greatly shortens the average waiting time of the UE.

6. Conclusion

In this paper, we propose an NB-IoT uplink data transmission optimization algorithm based on deep reinforcement learning. The algorithm considers the time-frequency domain resources of NB-IoT as the state space, the frequency domain position as the action, the neural network as the error function, and the resource utilization as the reward and punishment value. DDQN is designed to interact with the environment and iteratively train the algorithm model. Compared with the traditional algorithm, the simulation results have improved the resource utilization and the data transmission rate. Meanwhile, the average waiting time of the UE has also been greatly shortened. Therefore, this algorithm model can effectively solve the dynamic scheduling problem of NB-IoT under the circumstance of the data transmission for massive devices. In the future, we will introduce a software radio platform and embed algorithms into the platform. Use the combination of software and hardware to further verify the performance of the algorithm.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was sponsored by the Project of Cooperation between SZTU and Enterprise (nos. 2021010802015 and 20213108010030) and Experimental Equipment Development Foundation from SZTU (no. 20214027010032).

References

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low power wide area networks: an overview,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[2] A. Haridas, V. S. Rao, R. V. Prasad, and C. Sarkar, “Opportunities and challenges in using energy-harvesting for NB-IoT,” *SIGBED Review*, vol. 15, no. 5, pp. 7–13, 2018.

[3] Y. Miao, W. Li, D. Tian, M. S. Hossain, and M. F. Alhamid, “Narrowband internet of things: simulation and modeling,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2304–2314, 2018.

[4] A. Kumar Sultania, C. Delgado, and J. Famaey, “Implementation of NB-IoT power saving schemes in ns-3,” in *Proceedings of the 2019 Workshop on Next-Generation Wireless with ns-3, WNGW*, pp. 5–8, Association for Computing Machinery, New York, NY, USA, June 2019.

[5] R. Zhang, Z. Han, T. Zheng, and L. Ning, “Trajectory mining-based city-level mobility model for 5G NB-IoT networks,” *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5356193, 12 pages, 2021.

[6] W. Lu, Yu Ding, Y. Gao et al., “Resource and trajectory optimization for secure communications in dual-uav-mec systems,” *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[7] W. Lu, P. Si, G. Huang et al., “Swipt cooperative spectrum sharing for 6g-enabled cognitive iot network,” *IEEE Internet of Things Journal*, p. 1, 2020.

[8] L. Feltrin, G. Tsoukaneri, M. Condoluci et al., “Narrowband IoT: a survey on downlink and uplink perspectives,” *IEEE Wireless Communications*, vol. 26, no. 1, pp. 78–86, 2019.

[9] X. Liu and X. Zhang, “Rate and energy efficiency improvements for 5G-based IoT with simultaneous transfer,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5971—-5980, 2019.

[10] X. Liu, X. B. Zhai, W. Lu, and C. Wu, “QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2052–2061, 2021.

[11] I. M. Delgado-Luque, F. Blánquez-Casado, M. Garcia Fuertes et al., “Evaluation of latency-aware scheduling techniques for M2M traffic over LTE,” in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pp. 989–993, ISSN, Bucharest, Romania, August 2012.

[12] N. Afrin, J. Brown, and J. Y. Khan, “A delay sensitive LTE uplink packet scheduler for M2M traffic,” in *Proceedings of the 2013 IEEE Globecom Workshops (GC Wkshps)*, pp. 941–946, Atlanta, GA, USA, December 2013.

[13] A. Sampath, P. Sarath Kumar, and J. M. Holtzman, “On setting reverse link target SIR in a CDMA system,” in *Proceedings of the Technology in Motion 1997 IEEE 47th Vehicular Technology Conference*, pp. 929–933, ISSN, Phoenix, AZ, USA, May 1997.

[14] Q. Li, Y. Wu, S. Feng, P. Zhang, and Y. Zhou, “Cooperative uplink link adaptation in 3GPP LTE heterogeneous networks,” in *Proceedings of the 2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, pp. 1–5, ISSN, Dresden, Germany, June 2013.

[15] A. Durán, M. Toril, F. Ruiz, and A. Mendo, "Self-optimization algorithm for outer loop link adaptation in LTE," *IEEE Communications Letters*, vol. 19, no. 11, pp. 2005–2008, 2015.

[16] M. Gatnau Sarret, D. Catania, F. Frank et al., "Dynamic outer loop link adaptation for the 5G centimeter-wave concept," in *Proceedings of European Wireless 2015; 21th European Wireless Conference*, pp. 1–6, Budapest, Hungary, May 2015.

[17] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, "Redundant rule detection for software-defined networking," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 6, pp. 2735–2751, 2020.

[18] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, "Idle slots skipped mechanism based tag identification algorithm with enhanced collision detection," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2294–2309, 2020.

[19] S.-M. Oh and J.S. Shin, "An efficient small data transmission scheme in the 3GPP NB-IoT system," *IEEE Communications Letters*, vol. 21, no. 3, pp. 660–663, 2017.

[20] C.-W. Huang, S.-C. Tseng, P. Lin, and Y. Kawamoto, "Radio resource scheduling for narrowband internet of things systems: a performance study," *IEEE Network*, vol. 33, no. 3, pp. 108–115, 2019.

[21] M. Wu, W. Huang, K. Sun, and H. Zhang, "A DQN-based handover management for SDN-enabled ultra-dense networks," in *Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 2577–2465, ISSN, Victoria, Canada, November 2020.

[22] N. Yang, H. Zhang, and R. Berry, "Partially observable multi-agent deep reinforcement learning for cognitive resource management," in *Proceedings of the GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, Taipei, Taiwan, December 2020.

[23] Y. Zhang, X. Wang, and Y. Xu, "Energy-efficient resource allocation in uplink NOMA systems with deep reinforcement learning," in *Proceedings of the 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP*, pp. 2472–7628, ISSN, Xi'an, China, October 2019.

[24] Z. Han, R. Zhang, F. Jin, and L. Ning, "Optimization of NB-IoT uplink resource allocation via double deep Q-learning," in *In Proceedings of the 10th International Conference on Communications, Signal Processing, and Systems (CSPS)*, Chang Bai Shan, China, August 2021.

[25] S. Sinche, D. Raposo, N. Armando et al., "A survey of IoT management protocols and frameworks," *IEEE Commun. Survey. Tutorials*, vol. 22, no. 2, pp. 1168–1190, 2020.

[26] N. Yang, H. Zhang, K. Long, H. Hsieh, and J. Liu, "Deep neural network for resource management in NOMA networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 876–886, 2020.

[27] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440.