# Research on Random Access Control Strategy and Optimization Algorithm of Multi-type Terminals Based on Deep Reinforcement Learning

Shuhao Yuan, Zhi Yan(✉), Bo Ouyang, and Haoyong Duan

School of Electrical and Information Engineering, Hunan University, Changsha, China
yanzhi@hnu.edu.cn

**Abstract.** With the further development of 5G technology, large-scale machine-type communication technology has become the key to realize the interconnection of massive terminals. However, when massive terminals initiate the random access process at the same time, it will cause serious network congestion, especially in the application scenario where multiple types of terminals coexist. Severe network congestion will definitely affect the access delay and packet loss rate of delay-sensitive terminals. It is necessary to design a reasonable competition resolution mechanism to alleviate network congestion. Therefore, this paper proposes a random access optimization algorithm for multi-type terminals based on deep reinforcement learning. By introducing a priority design into the distributed queue access mechanism, the access opportunities of delay-sensitive terminals are expanded and the probability of collisions is reduced. An optimization algorithm based on the deep Q-learning network is proposed to dynamically adjust the number of preambles exclusively used by high-priority terminals, so as to reduce the influence of resource monopoly on the delay-tolerant terminals and minimizes conflicts as much as possible. In different load scenarios, the proposed algorithm is compared with existing competition resolution mechanisms and methods, and the practicability and effectiveness of the proposed method in solving the key problem of massive multi-type terminal coexistence are proved.

**Keywords:** Massive Machine-type Communication · Random Access · Deep Reinforcement Learning · DQN

## 1 Introduction

With the further development of the 5th mobile communication network (5G), 5G application scenarios are divided into enhanced mobile broadband (eMBB) scenarios, massive machine type communication (mMTC) scenarios, and low-latency and high-reliability (uRLLC) scenarios according to user's demand for communication technology. Among them, mMTC as the key technology to realize the Internet of Everything, can realize the communication between massive intelligent terminals without human intervention


F. Gao et al. (Eds.): Chinacom 2023, LNICST 590, pp. 3–20, 2024.
https://doi.org/10.1007/978-3-031-67162-3_1

[1]. The random access (RA) process is a basic and very important process in the wireless communication process. Its main functions include terminals initial access, uplink resources allocation, and uplink synchronization recovery [2]. In the traditional RA process, the terminal requests access by passing the preamble for identity authentication and synchronization to the base station, but when multiple terminals select the same preamble in the same RA cycle, resource conflict will occur. At this time, the terminals that have contention conflicts need to determine their retransmission time through a certain contention resolution mechanism. Therefore, it is important to design an efficient and reliable contention resolution control strategy and optimization algorithm, which is of profound significance for increasing the system capacity and promoting the further development of communication technology [3].

The research on the contention resolution mechanism in the RA process has been extensively discussed in the academic community. Through specific flow control methods, a large number of access requests are dispersed in the time domain as much as possible, so as to reduce the access congestion of the Physical Random Access Channel (PRACH) and the probability of terminal conflict, and reduce the possibility of congestion. Literature [4] proposes a queue-aware access control method based on the Access Class Barring (ACB) mechanism, which dynamically allocates the ACB factor for access control by sensing the length of the buffer queue of the device. Literature [5] proposes a sliding backoff window access control method to alleviate the network congestion caused by large-scale access, and dynamically adjusts the size of the backoff window by sensing the utilization rate of the data queue, which effectively improves the problem of packet loss caused by multiple retransmissions. Literature [6] applies the distributed queue mechanism to the LTE system, which reduced the probability of secondary collisions in the retransmission process by large-scale discrete grouping of conflicting devices in the time domain. It also proposes how to add queue information to the PRACH, which laid the foundation for the better application of the distributed queue random access mechanism to wireless communication technology. However, the existing random access contention resolution mechanism inevitably encounters problems such as high access delay and high packet loss rate when the number of requesting access devices is large. A new algorithm is needed to optimize the original access mechanism.

Since Deep Reinforcement Learning (DRL) has obvious advantages in resource allocation and decision optimization, it is an ideal tool for solving dynamic resource planning problems. Literature [7] uses a multi-agent deep Q-learning network with shared parameters to design a single conflict control strategy for all machine-type communication devices, so as to meet the differentiated needs of different types of devices and reduce signaling overhead. In reference [8], a back-off access scheme based on AHP-AC was designed. Firstly, the analytic hierarchy process (AHP) was used to obtain the Quality of Service (QoS) requirements from different types of users and group them, then through the Actor - Critic algorithm dynamically allocate back-off time slots in order to maximize the access success rate of devices while meeting the individual communication needs of devices. Literature [9] proposes an autonomous back-off access congestion control algorithm based on Q-learning. By solving the optimal access number and historical access conflict probability, the device can analyze the current environment before

accessing and actively make back-off decisions, which can effectively improve network energy efficiency and spectrum efficiency.

However, with the large-scale increase of machine-type communication devices, the access types and working environments are more complex. The effectiveness and optimization performance of the above solutions are greatly reduced. Therefore, it is necessary to design a solution that can adapt to large-scale multi-type data terminal access method of the scene. Based on the distributed queue mechanism, this paper proposes a priority distributed queue random access control strategy based on the deep reinforcement learning algorithm. The base station can allocate exclusive preamble resources for high-priority terminals according to the type of terminal requesting access, and dynamically adjust the number of exclusive preambles through learning experience. This method can meet the communication requirements of various types of data terminals in a large-scale multi-type data terminal access scenario. It reduces the average access delay and average energy consumption, and increases the system capacity.

## 2 System Model

### 2.1 Scenario Description

In this paper, we consider a single cell scenario where multiple types of data terminals coexist. As shown in Fig. 1, multiple data terminals are evenly distributed in the cell, which are divided into uRLLC type data terminals (UTDTs) and mMTC type data terminals (MTDTs). The system prioritizes terminals based on their sensitivity to delay.When the service arrives at the terminal for data transfer, the terminal may perform a four-step random access attempt as each RA opportunity (RAO) arrives. It is assumed that when the first RAO arrives, $M$ terminals receive the service data and initiate a RA process at the same time, where the number of UTDTs is $m$ and the number of MTDTs is $n$. The RAO contains $N$ preambles for data terminals competition access, and the remaining time slots are used for data transmission.

In a RAO period, in order to shorten the access delay of UTDTs as much as possible and reduce the data packet discarding caused by access blocking, UTDTs can select all preamble resources for random access process, while MTDTs can only share some preamble resources with some UTDTs. When multiple terminals select the same preamble to initiate an access request to the base station, it means that the terminals have access conflicts and cannot successfully access the network. When the next RAO arrives, the conflicting data terminals will continue to send access requests to the base station according to a certain contention resolution mechanism [10].

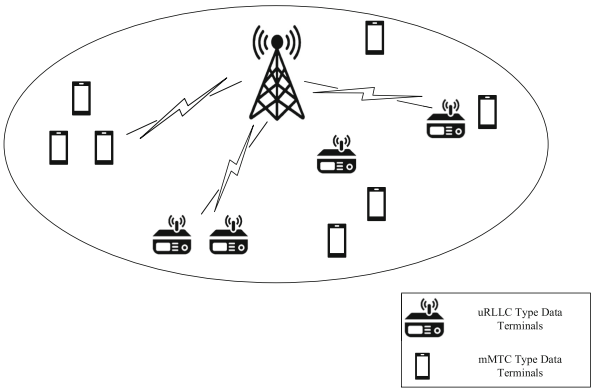


**Fig. 1.** Single-cell wireless communication system where multiple types of terminals coexist.

## 2.2 Priority-Based Distributed Queue Random Access Control Strategy

The distributed queue random access mechanism is often used in star topology network with a single coordinator or base station and a large number of data terminals. The basic idea is to introduce a contention queue to solve the conflict problem in the RA process. However, massive multiple types of terminals often coexist in many large-scale access application scenarios, so the base station or coordinator is required to allocate access resources reasonably based on the type and status of the terminal requesting access to ensure that the data delivery of UTDTs satisfies the basic requirements of low latency and high reliability application scenarios. Therefore, based on the traditional distributed queue mechanism, and a priority-based random access control strategy for distributed queues is proposed in this paper.

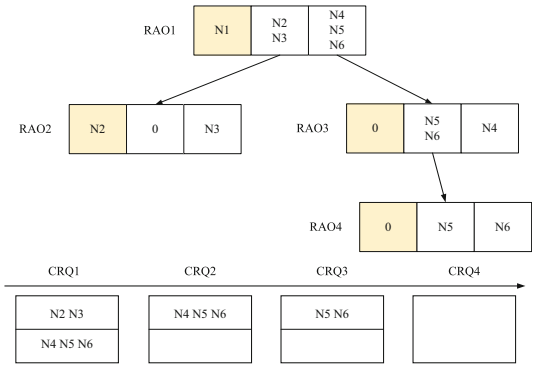


**Fig. 2.** Schematic diagram of priority-based distributed queue random access control strategy.

As shown in the Fig. 2, a total of 4 access opportunities are displayed from top to bottom, and they are sequenced in the time domain. Each long rectangle represents a RAO, and each RAO contains 3 random access resources. Among them, resource NO.1 is the exclusive access resource of UTDTs, and other resources are shared by UTDTs and MTDTs. Assuming that when the first RAO arrives, there are 6 terminals participating in the RA process, then an access resource is randomly selected to transfer to the base station in line with the priority principle. In the figure, two access resources have resource conflicts because they are selected by multiple data terminals. In this case, terminals that select the same resource are organized into a terminal group, and are added to the contention resolution queue (CRQ), when the next RAO arriving, the terminal group at the head of the CRQ queue is awakened and joins in the next RA process. In particular, when the number of terminal requests for access is greater than the maximum number of retransmissions tolerated by the system, the access request process will be regarded as a failure, thus the data packet will be discarded and the data terminal will enter a sleep state until the next transmission task arrives.

By introducing the above priority access idea, on the one hand, the access opportunities of UTDTs can be expanded and the access delay cost caused by resource conflicts can be reduced. On the other hand, because the terminal at the head of the CRQ queue can get the retransmission opportunity first in the following RAO, even if UTDTs have resource conflicts, it can also get the retransmission opportunity and sufficient access resources first in the following competitive resolution phase, which will reduce the probability of secondary conflicts and decrease the additional access delay cost and packet loss rate to a certain extent.

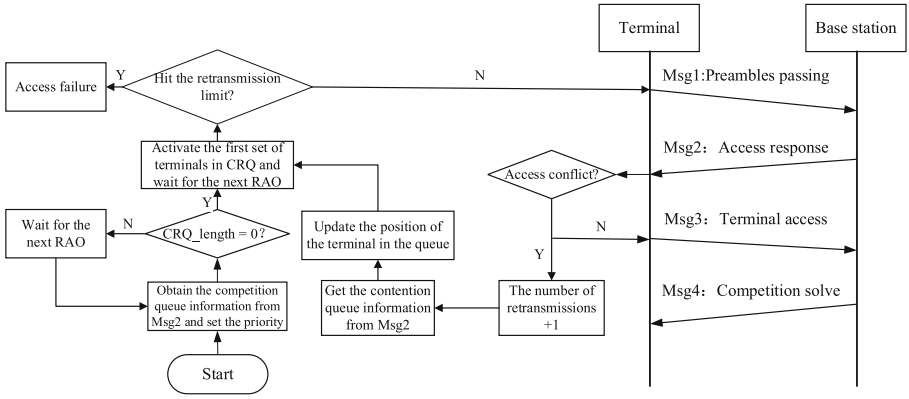


**Fig. 3.** The general access process of the priority-based distributed queue access mechanism

Figure 3 shows the complete access process of the priority-based distributed queue access mechanism. Before the terminal initiates an access request, it will listen to the SIB signal broadcast by the base station. The base station will inform the terminal of the preamble sequence and the idleness of the contention resolution queue. The random access process will continue only the contention queue is empty, otherwise the terminal will be suspended and wait for the RAO number of the current contention resolution

queue length before initiating an access request. After the terminal successfully selects the preamble sequence to transmit Msg1 to the base station, the base station will decode the received preamble and perform collision detection. Based on the distributed queue access mechanism, the base station will add access resource identifier *c_id* of the access resource that conflicts in this access cycle and contention queue length *CRQ_length* to the Msg2 feedback to the terminal. The terminal confirms whether it has a conflict and its relative position in the contention resolution queue according to the identification. Meanwhile, the terminal can judge its specific position in the queue according to the queue length [11].

## 3 Problem Description

As mentioned above, under the priority-based distributed queues random access mechanism, UTDTs can obtain more access opportunities and retransmission opportunities after conflicts. This mechanism can effectively reduce the conflict resolution delay and packet loss rate of UTDTs. However, with the continuous advancement of the RA process, if the proportion of UTDTs exclusively accessing resources is too large, it will inevitably lead to large-scale conflicts of MTDTs, resulting in a serious decline in the performance of the competition resolution queue handling conflicts. Therefore, further control measures are required for the priority-based distributed queue random access mechanism.

### 3.1 Delay Model

Assume that the number of preambles available in a single RA opportunity is $N$ and the number of terminals that initiate the RA process when the first access opportunity arrives is $M$. $\overline{R_{\max}}(M, N)$ represents the time slot length required for $M$ terminals to for all $N$ preamble resources to successfully execute the random access procedure. Suppose that in an RAO, $k$ of the $N$ preambles conflict, and $n_k$ terminal selects the $k$ th conflicting preamble resource. According to the distributed queue mechanism, the conflicting terminals will be divided into $k$ terminal groups, and complete random access attempts in the next $k$ RAOs until no multiple terminals select the same preamble resource for random access attempts. Therefore, based on the above analysis of the distributed queue random access process, the total time slot length required is as follows:

$$\overline{R_{\max}}(M, N) = 1 + \sum_{k=0}^{N} \sum_{n_1 \ldots n_k} P(k; n_1 \ldots n_k) \times (1 + \sum_{i=1}^{k} \overline{R_{\max}}(n_i, N)) \tag{1}$$

Among them, "1" indicates the first RAO, $\overline{R_{\max}}(n_i, N)$ indicates the sum of RAOs required by the conflicting terminal groups corresponding to the $k$ conflicting preambles in the first RAO to complete the random access process in the next RAO, $P(k; n_1 \ldots n_k)$ indicates the probability that $k$ access resources conflict, and there are $n_k$ terminals select the conflicting resource. The sum of all possible situations is the amount of RAO required for $M$ terminals to contend for $N$ preambles to complete the RA process, which is the total slot length required for contention resolution.

Furthermore, after the first RAO, the conflicting terminals are separated into $k$ terminal groups, which join the contention resolution queue and wait until the new RAO arrives to participate in the subsequent RA process. Therefore, in the subsequent access process, the terminal groups are independent of each other. Simplifying the repeated situation in the original model theory, the low-complexity theoretical model of the total time slot length required to complete the RA process is obtained as:

$$\overline{R_{\max}}(M, N) = 1 + \sum_{i=2}^{M} \frac{C_N^1 C_M^i (N-1)^{M-i}}{N^M} \times \overline{R_{\max}}(i, N) \tag{2}$$

Then the average number of retransmissions of the termina, that is, multiplied by the proportion of the terminal groups to the total number of terminals when calculating the total time slot [12, 13]:

$$\overline{R_{av}}(M, N) = 1 + \sum_{i=2}^{M} \frac{C_N^1 C_M^i (N-1)^{M-i}}{N^M} \times \overline{R_{\max}}(i, N) \times \frac{i}{M} \tag{3}$$

According to the priority-based distributed queue random access mechanism described in Sect. 2, combining the priority mechanism with the above delay model, it can be obtained that when the system allows UTDTs to monopolize p preambles, the time slot lengths required for $m$ UTDTs and $n$ MTDTs to complete the RA process are respectively:

$$T_U = \overline{R_{\max}}(\alpha m, p) + \overline{R_{\max}}((1-\alpha)m, (N-p)) \tag{4}$$

$$T_M = \overline{R_{\max}}((1-\alpha)m + n, (N-p)) \tag{5}$$

α is the probability of selecting an exclusive preamble in m UTDTs for random access process.

Therefore, the overall access delay of the system $T$ is:

$$T = T_U + T_M \tag{6}$$

### 3.2 Energy Consumption Model

For machine-type communication terminals, the main energy consumption comes from the signaling overhead of information interaction with the base station during uplink data transmission. The energy consumption for receiving downlink data and idle periods is very limited. Based on the distributed queue random access mechanism, terminals have four states during the RA process: idle state, listening state, backoff state and access state. When the terminal is idle before generating a demand for uplink data transmission, no energy consumption will be generated at this time. The terminal periodically monitors whether there is synchronization information from the base station. When the terminal has an uplink data transmission requirement, the terminal enters the listening state, and obtains the specific parameter configuration of this RA opportunity by listening

to the broadcast signal transmitting from the base station. At the same time, terminals listen to Msg 2 to obtain the terminal group in the current CRQ access status. If the terminal competes successfully during the access process, it will enter the access state and successfully uploads data packets to the base station. If the competition fails, it will enter the backoff state, that is, the terminal will queue up in the competition queue. Then the terminal will always be in an idle state until the next access attempt [14].

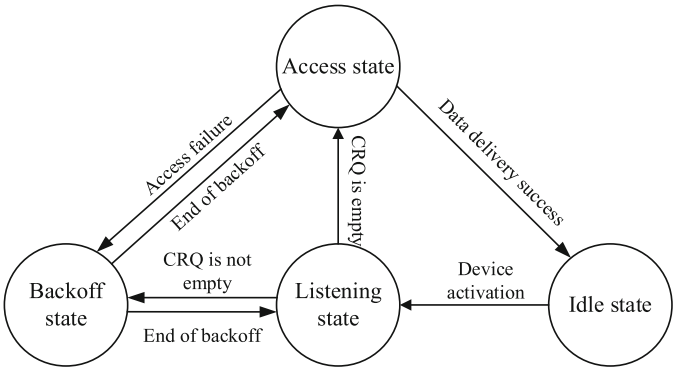


**Fig. 4.** Terminal access state transition diagram

Assuming that the number of terminals trying to access in a random access process is $M$, the number of UTDTs is $m$, the number of MTDTs is $n$, and the number of preambles is $N$. According to the priority-based distributed queue random access mechanism described above, the first RAO is regarded as the first layer for all terminals to complete the random access process, and the conflicting terminals are divided into several terminal groups to form the nodes of the following layer. So, the average access success rate of each type of terminal in the $i$ th layer can be calculated as:

$$P_U(i) = (1 - \frac{1}{N})^{\overline{m(i)}-1} \tag{7}$$

$$P_M(i) = (1 - \frac{1}{(1-p)N})^{\overline{n(i)}-1} \tag{8}$$

Among them, $p$ represents the proportion of the preamble exclusively owned by UTDTs, $\overline{m(i)}$ represents the average number of UTDTs in the $i$ th layer, and $\overline{n(i)}$ represents the average number of MTDTs in the $i$ th layer. Afterwards, the average number of each type terminals participating in each layer can be calculated from the number of conflicting terminals in the previous layer:

$$\overline{m(i)} = \frac{[1 - P_U(i-1)] \cdot \overline{m(i-1)}}{G(i-1)} \tag{9}$$

$$\overline{n(i)} = \frac{[1 - P_M(i-1)] \cdot \overline{n(i-1)}}{G(i-1)} \tag{10}$$

wherein, $\overline{m(i-1)}$ is the average number of UTDTs conflicts in the ($i$-1) th layer nodes; $\overline{n(i-1)}$ is the average number of MTDTs conflicts in the i-1th layer nodes; $G(i-1)$ represents the discrete number of the ($i$-1) th layer terminal group, and $G(i-1) = \overline{m(i-1)} + \overline{n(i-1)}$, which can be obtained from the contention queue information transmitted downlink by the base station. Therefore, the number of nodes in the $i$ th layer can be calculated as:

$$R(i) = \begin{cases} 1 & i = 1 \\ R(i-1) \cdot G(i-1), i > 1 \end{cases} \tag{11}$$

Assuming that the last layer is $i_f$, then when $[1 - P(i_f - 1)] \cdot \overline{M(i_f - 1)} < 1$, there will be no new conflicting terminals, and the average number of times to enter the frame listening state can be obtained:

$$\overline{K} = \frac{\sum_{i=2}^{i_f} R(i) \cdot (i_f - i + 1)}{\sum_{i=2}^{i_f} R(i)} \tag{12}$$

Therefore, the average energy consumption of the terminal processing the RA process can be calculated as:

$$\begin{aligned} E = \sum_j \big[ & (\overline{R_{\max}}(m_j, N) + \overline{R_{\max}}(n_j, N-p)) \cdot E_B \\ & + (\overline{R_{av}}(m_j, N) + \overline{R_{av}}(n_j, N-p)) \cdot (E_A - E_B) \\ & + (\overline{K}(m_j, N) + \overline{K}(n_j, N-p)) \cdot (E_H - E_B) \big] \cdot \frac{m_j + n_j}{M} \end{aligned} \tag{13}$$

Among them, $E_B$, $E_H$, $E_A$ is the energy consumption of the terminal in the back-off state, the listening state, and the access state.

### 3.3 Optimization Problem

In order to minimize the access delay and energy consumption of data terminals in the RA process, this paper studies the strategy optimization under the priority-based distributed queue random access mechanism. It can be expressed as:

$$\min_p \quad T_{average}, E_{average} \tag{14}$$

$$\text{s.t.} \quad 0 \le p_{exclusive} \le N \tag{15}$$

$$T_U \le T_{\max} \tag{16}$$

$$R_{U\_av} < R_{\max} \tag{17}$$

$$R_{M\_av} < R_{\max} \tag{18}$$

Among them, formula (15) is the constraint on the number of UTDTs exclusive preambles, formula (16) is the constraint on the delay of UTDTs disposing random access process and the $T_{\max}$ denotes the maximum tolerance experiment of UTDTs, formula (17) and formula (18) are the constraints on the number of retransmissions after UTDTs conflicts and the constraint on the number of retransmissions after MTDTs conflicts and the $R_{\max}$ denotes the maximum number of retransmissions tolerated by the system.

The transformation of data terminals between various types of states has strong randomness, and the probability of state transition is determined by the results of terminal access in the previous stage. Therefore, the process of using distributed queues to dispose random access can be fitted as a Markov decision process. Deep reinforcement learning can obtain the optimal strategy without using complex mathematical analysis methods, and has a good processing ability for the strategy optimization problem of the priority-based distributed queue random access mechanism in complex situations.

## 4 Multi-type Terminals Random Access Optimization Algorithm Based on DQN

### 4.1 DQN Algorithm Framework and Neural Network

The Deep Q-Learning Network (DQN) algorithm is a classic different-strategy temporal difference algorithm in the deep reinforcement learning algorithm. It uses the neural network approximation function to learn the optimal strategy, and is often used in low-dimensional discrete action spaces optimization problem. This paper proposes to realize the dynamic division of access resources in the priority distributed queue random access mechanism based on DQN, so as to solve the above optimization problems.

The DQN algorithm framework is shown in Fig. 5, which is mainly composed of agents, environments, experience playback pools, loss functions, value networks and target networks. As the Q-Learning algorithm is an off-policy algorithm, by introducing the experience playback mechanism, the experience samples of the agent at each moment are stored in the experience playback pool, which can reduce the correlation between experiences and make the neural network convergence is more efficient. In addition, the DQN algorithm calculates the Q value through two neural networks with the same structure but different parameters, in which the value network is used to calculate the Q value of strategy selection and iterative update, and the target network is used to calculate the Q value of the next state in the TD target, which can keep the training process stable and speed up the convergence.

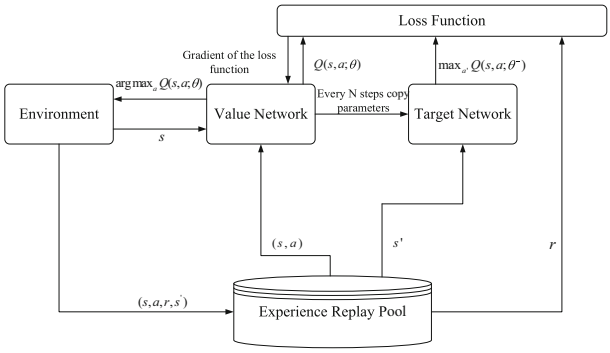


**Fig. 5.** DQN algorithm structure

First, the algorithm randomly selects an initial state $s_t$ and selects an action $a_t$ with a ε-greedy strategy based on the value of the current state. Then the agent acts on the environment, gets a reward $r_t$ and the next state $s_{t+1}$. The resulting state transformation quadruple $[s_t, a_t, r_t, s_{t+1}]$, which contains states, actions, rewards, and next-moment states, is stored in the experience return pool. At the beginning of each training, the Q value of the state will be updated and the update formula of the Q value is:

$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + l\Big[R_{t+1} + \gamma \max_a Q_k(s_{t+1}, a) - Q_k(s_t, a)\Big] \tag{19}$$

where, l is the learning rate, $R_{t+1}$ is the reward obtained from the experience playback pool, and $\gamma$ is the discount factor. When the capacity in the experience playback pool is large enough, a certain number of state transitions are randomly extracted from it and calculate the target Q value $Q_{target}$ of the current state:

$$Q_{target} = R_i + \gamma \max_{a_i} Q_i(s_i, a_i, \theta^-) \tag{20}$$

Among them, $a_i$ is the action that maximizes the Q value of the target network, and $\theta^-$ is the parameter of the target network. Then use the loss function to calculate the mean square error cost between the target network and the value network:

$$Loss(\theta) = \frac{1}{2}[Q_{target} - Q(s_i, a_i; \theta)]^2 \tag{21}$$

Then DQN updates the network parameters $\theta$ through the gradient descent method, and copies the parameters to the target network after every certain length of steps. The gradient descent formula is as follows:

$$\theta_{t+1} = \theta_t + E\big[Q_{target} - Q(s_t, a_t; \theta_t)\big]\nabla Q(s_t, a_t; \theta_t) \tag{22}$$

### 4.2 Basic Flow of Multi-type Terminals Random Access Optimization Algorithm Based on DQN

In order to realize the RA processing of data terminals and minimize energy consumption, the state space, action space and reward function of DRL are designed based on the system model established in this paper, as follows:

- State space: In the process of data terminals implementing the priority-based distributed queue random access mechanism, the delay of the random access process is related to the number of various terminals, the number of preambles exclusive to UTDTs and the length of the contention resolution queue. The expression of the state space is: $s(t) = \left[U(t), M(t), P, L_{CRQ}\right]$, where $U(t)$ and $M(t)$ represent the number of UTDTs and MTDTs participating in this RAO, P denotes the number of exclusive preambles of UTDTs, and $L_{CRQ}$ represents the length of the current contention queue.
- Action space: In order to reduce the processing delay and energy consumption in the access process and ensure the access priority of UTDTs, we will dynamically adjust the number of exclusive preamble resource for ensuring the priority access of UTDTs and the access success rate of MTDTs in this RAO is improved under the premise of priority. The expression of the action space is: $a(t) = [P-2, P-1, P, P+1, P+2]$.
- Reward function: In a RAO, the main evaluation indicators affecting the delay and energy consumption of the access process include the access success rate of UTDTs and MTDTs, the length variation of CRQ and the number of lost packets. In order to ensure the access delay and energy consumption, and satisfy the constraints of the access delay and retransmission times of UTDTs and MTDTs in Eqs. (16), (17) and (18), the reward $r_{usp}$ and $r_{msp}$ are designed. In addition, when the length of the contention resolution queue is longer after one RAO, that is, the resource conflict is more intense, a penalty $r_{CRQ}$ is set. When a packet loss occurs in this RAO, set a penalty $r_{loss}$. So the expression of the reward function is: $r(t) = r_{usp} + r_{msp} + r_{CRQ} + r_{loss}$.

Based on the above design, the RA optimization algorithm process of the priority distributed queue based on DQN proposed in this paper is shown in Algorithm 1 (Table 1):

**Table 1.** The training and learning process of optimization algorithm

| **Algorithm 1** Multi-type Terminals Random Access Optimization Algorithm Based on DQN |
|---|
| 1: Initialize the parameters of DQN network and RA process |
| 2: Initialize the state s(t) through the message passing network; |
| 3: Loop |
| 4: for i = 1 : 1 : epoch |
| 5: The base station selects an action a(t) according to the ε-greedy strategy and reserves part of the preambles for UTDTs |
| 6: Calculate the reward after performing the RA process, and update the CQR |
| 7: Get the next state s(t+1) |
| 8: Put the experience data (s(t), a(t), r, s(t+1)) into the experience playback pool |
| 9: Calculate the loss function and update the target network parameters |
| 10: Minimize the loss function, update the network parameters $\theta$ |
| 11: End the loop |

## 5 Simulation Results

In this paper, a personal computer is used to verify the performance of the proposed optimization algorithm. The simulated software environment is Python 3.6 and the simulated hardware platform is personal computer with Intel Core i5-12500H 2.50 GHz processor and 16 GB memory. Assuming that 500 terminals are uniformly distributed in a single base station cell with a radius of 500 m, among which there are 200 UTDTs and 300 MTDTs. In one RAO, the terminals perform the RA process by competing for 20 preamble sequences and the maximum number of retransmissions tolerated by the system is 7. It is assumed that the system simulation adopts the collision channel model, that is, there is no transmission error caused by the imperfect wireless channel in the process of signaling interaction and data transmission between the terminal and the base station. Other parameter designs about the optimization algorithm are shown in Table 2.

**Table 2.** DQN algorithm design parameters

| Parameter | Value |
|---|---|
| Number of training rounds | 5000 |
| Learning rate α | 0.0001 |
| Discount factor gamma | 0.99 |
| Experience playback pool length | 5 000 |
| Parameter update cycle | 150 steps |
| Network layers | 2 |
| Max ε | 0.9 |

Simulation results are provided below to verify the performance of the proposed DQL based random access optimization algorithm for priority distributed queues. In the results, refer to the scheme as "DQN –PADQ".

Figure 6 describes the convergence of the DQN-PADQ algorithm, where the abscissa is the number of model training times and the ordinate is the value of the model loss function. It can be seen that as the number of trainings increases, the loss function value gradually approaches the local optimal solution. When the number of trainings is close to 175,000 times, the model basically converges. Figure 7 shows the changes in the delay, number of retransmissions, and energy consumption of the data terminal as the number of iterations increases, where the abscissa is the number of model training times and the ordinate is the reward function value of the model, system access delay, the average number of retransmissions and system average energy consumption. It can be seen that with the number of training increases, the performance of key indicators in all aspects of the system gradually tends to the local optimal solution.

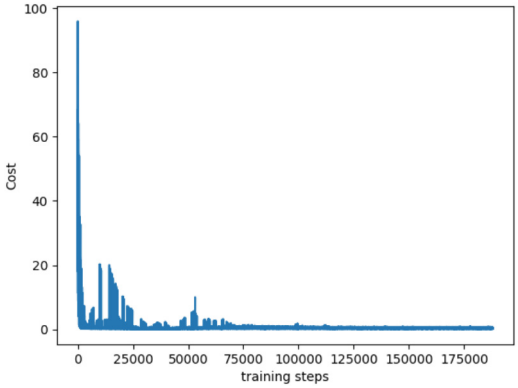


**Fig. 6.** The relationship between the number of training times and the value of the loss function

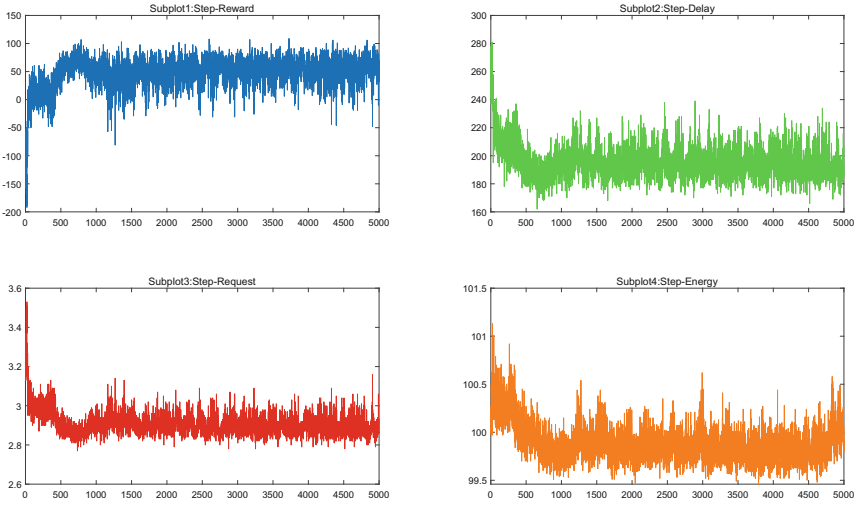


**Fig. 7.** Relationship between training times and performance indicators

In addition, in order to show the optimization performance of DQN-PADQ algorithm, this paper also considers the key performance comparisons between DQN-PADQ algorithm and ACB mechanism, Back-off mechanism, DQ mechanism and PSO-DQ algorithm. The ACB mechanism incorporates a limiting factor to restrict the number of access requests during each RAO arrival. By implementing a backoff window, the collision terminal is granted the ability to select a specific time within the window for initiating subsequent access attempts. DQ mechanism serves as an inherent distributed queue mechanism that categorizes colliding terminals into groups and enqueues them in contention resolution queues, resolving conflicts upon subsequent RAO arrivals. The PSO-DQ algorithm combines particle swarm optimization with the distributed queue mechanism to determine an optimal grouping method and minimize average system delays.

Figure 8 shows a comparison chart of the terminals average access delay under different access mechanisms and optimization algorithms, where the abscissa is the number of terminals and the ordinate is the average access delay. When the number of terminals is at a small level, the delay of the access process under the five access mechanisms has little difference. When the number of terminals continues to increase, the access delay of the five access mechanisms is significantly improved. But the growth of ACB or Back-off mechanism is more prominent, while the growth of the average delay of the three access mechanisms based on distributed queues is relatively stable, which is because distributed queue discretized the conflict terminals in the time domain by the queue-type queue. The probability of secondary collisions of the terminals is reduced, so the average delay increases relatively slowly. At the same time, comparing three access mechanisms and optimization algorithms based on distributed queues, it is obvious that the DQN-PADQ algorithm performs better in delay optimization. By dynamically adjusting the number of preambles exclusive to UTDTs, it can not only guarantee the delay and reliability requirements of UTDTs, but greatly alleviate the access conflict of MTDTs. The above results show that when the number of data terminals requesting access is large, the DQN-PADQ algorithm can reduce the average access delay more effectively.

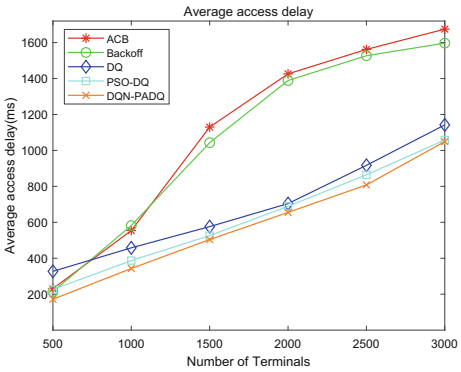


**Fig. 8.** Comparison of average access delay

Figure 9 shows the comparison of the average number of terminal requests under different access mechanisms and optimization algorithms, which reflects the ability to handle congestion to a certain extent, where the horizontal coordinate is the number of terminals and the vertical coordinate is the average number of requests. As shown in the figure, with the increasing number of requests for access, the average number of terminals requests under the five access mechanisms has an upward trend. But the average number of requests for terminals under the ACB or Back-off mechanism increases significantly, even approaching the maximum number of retransmissions tolerated by the system. The average number of terminal requests increases relatively stable under the access mechanism and optimization algorithm based on distributed queues. The distributed queue mechanism effectively partitions conflicting terminals into discrete groups during the first RAO, resulting in a significant reduction in the number of participating terminals

during subsequent RAOs. Simultaneously, prioritization constraints enable UTDTs to not only have priority for secondary access but also provide them with a wider range of preamble resources to choose from. Consequently, the system experiences fewer average access requests. In comparison, the DQN-PADQ algorithm can keep the average number of requests is kept at about 3.5 times when 3000 terminals initiate access at the same time. It shows that the DQN-PADQ algorithm can more effectively reduce the average number of access requests of terminals in the access process.

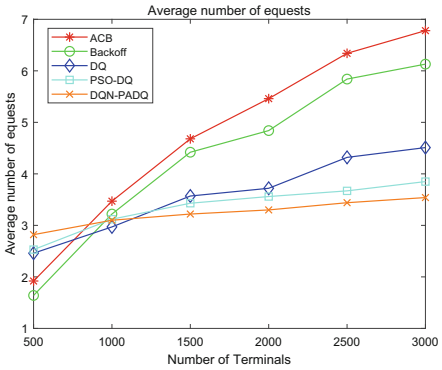


**Fig. 9.** Comparison of average number of access requests

Figure 10 shows a comparison of the average energy consumption of terminals under different access mechanisms and optimization algorithms, where the abscissa is the number of terminals and the ordinate is the average energy consumption. Obviously, as the number of access requests continues to increase, the average energy consumption of terminals under all access mechanisms increases. However, in scenarios with different numbers of terminals, the average energy consumption of terminals under the ACB or Back-off mechanism is always higher than that of the other three comparison schemes. The terminal's access state transition diagram in Fig. 4 reveals that the primary source of additional energy consumption during the access process is the transition between the backoff state and the listening state. However, due to the significant reduction in average terminal access requests achieved by implementing a distributed queue mechanism, this extra energy consumption is minimized, resulting in lower average system access energy consumption. Among the three access mechanisms and optimization algorithms based on distributed queue, the optimization effect of the DQN-PADQ algorithm is obviously better than the other two schemes. The above results prove the superiority of the DQN-PADQ algorithm in optimizing the average energy consumption in the access process.

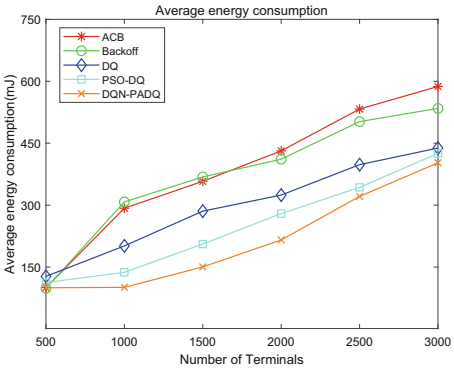


**Fig. 10.** Comparison of average access energy consumption

## 6 Conclusion

This paper considers the large-scale random access process of data terminals, and the goal is to reduce the average access delay and access energy consumption in a large number of coexistence scenarios of multi-types terminals through the priority design. In order to solve the problem that excessive resource monopoly affects the access performance of MTDTs, which affects the overall system performance during the access process, this paper proposes a priority distributed queue random access algorithm based on DRL. The simulation results show that the control strategy and its optimization algorithm proposed in this paper can effectively reduce the average access delay and energy consumption of the whole system, and its performance is more prominent than the same type of algorithms.

## References

1. Jiangfeng, C., Weihai, C., Fei, T., Chun-Liang, L.: Industrial IoT in 5G environment towards smart manufacturing. J. Ind. Inf. Integr. **10**, 10–19 (2018)
2. Choi, H., Moon, H.: Simulation on delay of several random access schemes. In: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 149–151 (2019)
3. Miuccio, L., Panno, D., Riolo, S.: Joint control of random access and dynamic uplink resource dimensioning for massive MTC in 5G NR based on SCMA. IEEE Internet Things J. **7**(6), 5042–5063 (2020)
4. Chowdhury, M.R., De, S.: Queue-aware access prioritization for massive machine-type communication. IEEE Internet Things J. **9**(17), 15858–15873 (2022)
5. Li, Y., Lv, Z., Fan, Z., Zhang, H.: Adaptive two-step binary exponential backoff strategy for random access. In: 2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 1104–1109 (2023)
6. Laya, A., Alonso, L., Alonso-Zarate, J.: Contention resolution queues for massive machine type communications in LTE. In: 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 2314–2318 (2015)

7. Jadoon, M.A., Pastore, A., Navarro, M.: Collision resolution with deep reinforcement learning for random access in machine-type communication. In: 2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring), pp. 1–6 (2022)
8. Mingtong, X.: Research on random access scheme for large-scale machine type communication with different QoS requirements. Master, Jilin University (2022)
9. Wu, X.: Research on MTC Random Access and Heterogeneous Network Resource Allocation Method Based on Reinforcement Learning. Master, Nanjing University of Posts and Telecommunications (2022)
10. Chen, Y., Wang, G., Yi, H., Zhang, W.: Priority-based distributed queuing random access mechanism for mMTC/uRLLC terminals coexistence. In: 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), pp. 1–6 (2021)
11. Shuchen, X., Xiangming, W., Zhaoming, L., Qi, P., Wenpeng, J.: Performance analysis and enhancement of random access process in cellular-IoT. J. China Univ. Posts Telecommun. **26**(6), 1–10 (2019)
12. Han, B., Schotten, H.D.: Grouping-based random access collision control for massive machine-type communication. In: GLOBECOM 2017–2017 IEEE Global Communications Conference, pp. 1–7 (2017)
13. Cheng, R.G., Becvar, Z., Yang, P.H.: Modeling of distributed queueing-based random access for machine type communications in mobile networks. IEEE Commun. Lett. **22**(1), 129–132 (2017)
14. Vázquez-Gallego, F., Alonso-Zárate, J., Tuset-Peiro, P., Alonso, L.: Energy analysis of a contention tree-based access protocol for machine-to-machine networks with idle-to-saturation traffic transitions. In: 2014 IEEE International Conference on Communications (ICC), pp. 1094–1099 (2014)
15. Tsoukaneri, G., Wu, S., Wang, Y.: Probabilistic preamble selection with reinforcement learning for massive machine type communication (MTC) devices. In: 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1–6 (2019)
16. Pacheco-Paramo, D., Tello-Oquendo, L.: Delay-aware dynamic access control for mMTC in wireless networks using deep reinforcement learning. Comput. Netw. **182**, 107493 (2020)
17. Yuan, W., Zang, Y., Zhang, L.: Preamble selection and allocation algorithm based on Q learning. In: Sun, X., Zhang, X., Xia, Z., Bertino, E. (eds.) Artificial Intelligence and Security. ICAIS 2021. LNCS, vol. 12736, pp. 443–454. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78609-0_38