ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20220225044110
Original scientific paper

Layout and Location of Water IoT Device Based on Few-Shot Reinforcement Learning

Shikai XING

Abstract: After the traditional water equipment integrates the communication module, IoT (Internet of Things) device is formed. Whether these battery-powered IoT devices can be installed in a certain location depends on whether the power consumption of these IoT devices in these locations can meet the expected life cycle. In this paper, by adopting strategies to save the power consumption of IoT devices when sending data, more locations can be selected to install IoT devices. The process of IoT device sending data packet sequence needs to be aware of the environment, interact with the environment, then make a decision, and then adjust the policy according to the effect of the action. Therefore, in this paper, the process of IoT device sending data packet sequence is modelled as MDP (Markov Sequence Decision Process), and the real-time SINR of channel and the transmission delay of data packet sequence are defined as the state space, and the action space consists of immediate transmission and delayed transmission, with the minimum total power consumption as the objective function. Because IoT devices are very sensitive to power consumption and cannot collect a large amount of data for training, this paper uses the Proximal Policy Optimization algorithm based on prior distribution to conduct few-shot reinforcement learning to quickly obtain the optimal decision sequence of layout and location of IoT devices.

Keywords: few-shot learning; layout and location; MDP; optimal power consumption

1 INTRODUCTION

Different from solving the game problem among multiple devices through reinforcement learning, solving the decision problem of single device through reinforcement learning can be modelled as MDP (Markov Decision Process) [1]. When the state transition probability of a single device is known, the dynamic programming iterative equation can be used to solve its decision-making problem. Because the wireless environment of the installation location of water device changes randomly, the state transition probability is often unknown, and reinforcement learning algorithm can explore the environment through trial and error, and obtain a good policy to maximize the overall expected reward. In this paper, a few-shot reinforcement learning algorithm based on prior distribution will be used to solve the Markov sequence decision model problem of single intelligent device.

2 LITERATURE REVIEW

In recent years, wireless communication technology is developing rapidly, especially the Internet of Things (Internet of Things) communication technology represented by NB-IoT (Narrowband IoT) network [2], which has the characteristics of deep coverage, low power consumption and low speed, and is very suitable for the wireless environment, data transmission and battery power supply of water device. With the comprehensive coverage of NB-IoT network, urban water supply and drainage pipe network and meter reading system begin to install a large number of IoT devices or install communication module on devices without networking, which makes the number of IoT devices such as smart water meters and smart manhole covers with communication module increasing rapidly.

According to the protocol between IoT devices and platform, the sensors of IoT device collect data according to certain rules and frequencies [3]. The amount of data accumulated in each time period is basically fixed, and these data are stored in IoT devices and uniformly sent to the platform [4]. The IoT device uploads the accumulated data to the platform in increments or absolute quantities at a predetermined time every day. The accumulated data before uploading will be split into data packet sequences and then sent [5].

The installation location of intelligent water device has a relatively similar environment, mainly installed in corridors, hoistways and other areas with weak wireless signal coverage [6]. In these areas, there will be random disturbances and noise superposition caused by various frequency shifts, interference, such as white noise, occlusion, co-channel interference, etc., and the channel quality will change randomly as a result of these interference superpositions [7]. Therefore, the channel quality of wireless communication can be jointly determined by signal coverage strength and noise interference, that is, SINR (signal to interference plus noise ratio) [8].

The IoT devices of urban pipe network system or Meter reading system are sensitive to battery power supply and power consumption, therefore, the data that the IoT device can send is limited [9]. Due to the occlusion reflection or stray interference of buildings, the wireless signal will be greatly attenuated, hence the battery life is very dependent on the installation environment and data transmission policy [10]. If the installation location and sending policy are not good, the battery will be quickly consumed after the IoT device is started, which cannot support the data collection and transmission business requirements with a period of 6 - 10 years [11]. Because the installed device is unattended, when the device needs to replace the battery, it is generally necessary to manually go to the site for operation [12]. When the battery is replaced manually on the spot or the traditional manual meter reading is adopted, it is often difficult to enter the house and the device maintenance cost is relatively high [13].

Because of the high labour cost of replacing batteries in these devices, considering from the investment and cost, it is expected that these devices can maintain a longer life cycle under the condition of sending the same amount of data, so the policy problem of which devices can be equipped with communication module arises.

The location of device with communication module depends on two aspects: the wireless signal environment of the device installation point and the device's perception of the environment [14]. The stronger the device's perception of the installation environment, the lower the requirement of wireless signal coverage quality and vice versa. By improving the device's ability to perceive the environment, the requirements of the quality of signal coverage at the installation site can be reduced, so that more sites have the conditions to install the IoT device.

When the channel state changes, the SINR will change, which will affect the power consumption of the IoT device when sending data. When the signal gets worse, it takes longer to send data, which requires more power consumption. When the signal quality becomes worse, the power consumption increases slowly; but when the signal quality is lower than a certain inflection point, the power consumption begins to increase rapidly, and the relationship between power consumption and signal quality presents a nonlinear relationship.

3 PROBLEM DESCRIPTION

Based on the above nonlinear relationship between channel quality and power consumption, there is no clear action judgment value. Under the condition that the battery capacity of the IoT device is unchanged, it is necessary to decide on the best action value according to the environment and a certain policy. That is, the IoT device detects the quality of the wireless channel through a sensor to determine whether to send data immediately or suspend sending data until the signal quality improves using a certain policy to achieve the minimum total power consumption and the longest service life of the IoT device after sending all the data of the day. In this paper, taking 6 - 10 years as the life cycle goal of the device, according to the amount of data to be sent, the location policy of whether the device can be equipped with communication module is determined.

Therefore, the layout and location of the IoT device and few-shot learning are essentially the problems that intelligent device can quickly converge by collecting a small number of task samples in an unknown environment. It is an optimization problem to realize the minimum power consumption of device, which belongs to the category of system optimization and intelligent decision-making.

4 SYSTEM MODEL

In this section, the environmental perception and decision-making process of the data packet sequence sent by the IoT device is modelled as MDP (Markov decision process) to describe the state transition of the system, and based on this process, the layout and location mechanism of the intelligent water IoTdevice is deduced [15].

The data to be sent by the IoT device at a time is composed of n data packets, and these data packet sequences form a set , $T_i = \{t_1, t_2, ..., t_n\}$, which constitutes a sample of task distribution. Each task sample corresponds to an MDP process, which is defined by a Quintuple, $\{S, A, P, R, \gamma\}$. Among them, S is the state space, A is the action space, $\boldsymbol{P}$ is the state transfer matrix, $\boldsymbol{R}$ is the state transfer reward, and γ is the long-term reward discount.

4.1 Channel State

The transmission device sends each data packet in turn, and according to the protocol, the amount of data sent every day is roughly the same. Due to the random characteristics of wireless channels, the channel quality will change randomly. When the channel quality is poor, it may lead to higher bit error rate or loss of some data packets, so it is necessary to resend these data packets.

SINR of the channel obeys small-scale Rayleigh distribution, and its probability density function [16] is expressed as:

$$p\left(snr\right) = \frac{1}{\overline{snr}} \exp\left(-\frac{snr}{\overline{snr}}\right) \tag{1}$$

Among them, snr represents a set of threshold values of SINR, and $\left(\overline{snr}\right)$ represents the expectation of SINR. Set a total of $N - 1$ channel quality threshold values of NB-IoT network, $snr = \{snr_1, snr_2, \ldots, snr_{(N\text{-}1)}\}$. According to snr, the channel quality can be divided into N states, $C = \{c_0, c_1, \ldots, c_{(N\text{-}1)}\}$.

Channel state probability is:

$$p_c\left(c_i\right) = \int_{snr_i}^{snr_{i+1}} p\left(snr\right) d\left(snr\right) \tag{2}$$

Assuming that the next moment of the channel can only be transferred to the adjacent interval, the probability of state transition of the channel is

$$p_c\left(c_i, c_{i+1}\right) = \frac{f\left(snr_{i+1}\right) \cdot T_f}{p_c\left(c_i\right)}, \, i \in \{0, 1, ..., N-2\} \tag{3}$$

$$p_c\left(c_i, c_{i-1}\right) = \frac{f\left(snr_i\right) \cdot T_f}{p_c\left(c_i\right)}, \, i \in \{1, 2, ..., N-1\} \tag{4}$$

Among them, $f\left(snr_i\right) = \sqrt{\frac{2 \cdot \pi\left(snr_i\right)}{\overline{snr}}} \cdot f_D \cdot exp\left(-\frac{snr}{\overline{snr}}\right)$, f_D is the maximum Doppler shift.

4.2 DeviceTime Delay State

The second component of state of the device is described by the total time delay after the device sends the ith data packet. Let the sending time of the first data packet be t_0, and the sending time of the ith data packet be t_i. Obviously, the total delay after the device sends the i-th data packet is not only related to the real-time decision of the current data packet, but also depends on the delay accumulation of the first $i - 1$ data packets. The total delay after the i-th packet is sent can be transformed into the combination of the current packet delay and the total delay of the previous $i - 1$ packets.

$$\tau_i = \tau_{i-1} + \Delta\tau_i + T_f \tag{5}$$

where τ_i is the delay state of the first i packets, and $\tau_{(i-1)}$ is the delay state of the first $i-1$ packets. $\Delta\tau_i$ is the waiting time determined by current decision, which is related to the current channel state. T_f is a constant, which is the sending time of a data packet. The device delay state set can be expressed as $\tau = \{\tau_1, \ldots, \tau_n\}$.

4.3 System Status and Action

System state S can be defined as the combination of channel state and device delay state, that is, $S \triangleq C \otimes \tau$. The above Eq. (5) is a recursive process. Because the transition probability is unknown, it cannot be solved by dynamic programming method.

The action taken by the IoT device at the moment when the new data packet is to be sent can be expressed as $a \in A, A \triangleq \{0, 1\}$. When $a = 0$, it means to send immediately; when $a = 0$, it means that the transmission is delayed.

4.4 System State Transition

When the IoT device sends a data packet, it may choose the policy of sending immediately according to the channel state. Based on the instantaneous SINR of the channel, the policy makes the device work in the connected state and then shifts to the new channel state and device delay state after the action.

The IoT device may also adopt the policy of delaying sending a data packet, which makes the device work in idle state and expects to transmit again when the channel state is better. After a certain interval time $\Delta\tau_i$, the IoT device may choose to continue transmitting according to the change of the channel state. After the implementation of this policy, the system state is transferred to the new channel state and device delay state. Because each packet is relatively small, the time for the device to send a packet is very short compared with the waiting time.

Let τ_i be the total delay of the first i data packets, and $\Delta\tau_i$ be the single-step delay of the ith data packet, that is, the waiting time of single-step pause. This waiting time is randomly determined by the IoT device according to the environment, which will delay the whole time of sending data packets. Let t_i be the time when the *i*-th data packet is ready to be sent, then the time delay before the $(i-1)$th data packet is ready to be sent is $t_{(i-1)} - t_0$. After the $(i-1)$th packet executes the action, the device state changes to.

After the ith packet decision, the recursive formula of the device delay state is

$$\tau_{i-1} = \begin{cases} t_{i-1} - t_0 + T_f, \text{ send immediately} \\ t_{i-1} - t_0 + \Delta\tau_{i-1} + T_f, \text{ send delayed} \end{cases} \tag{6}$$

$$\tau_{i-1} = \begin{cases} t_{i-1} + T_f, \text{ send immediately} \\ t_{i-1} + \Delta\tau_i + T_f, \text{ send delayed} \end{cases} \tag{7}$$

Let $\tau_0 = 0$, then the single-step delay of the ith data packet is

$$\Delta\tau_i = \left\{ int\left[N - snr_i \middle/ \left(\frac{snr_N - snr_1}{N} \right) \right] \right\} \cdot \delta \tag{8}$$

Time delay is related to snr, and the better snr, the smaller the time delay. Let δ be the parameter of delay time micro-segment, then in this paper, it is assumed that the channel quality in the micro-segment remains unchanged and the channel is irrelevant when retransmitting. Nis the number of states, and the function intis a rounding operation.

4.5 System Benefits and Costs

When the IoT device sends a data packet, if it does not receive the confirmation message from the platform within a certain period of time, it will resend the data packet. In order to avoid the difference of power consumption when retransmitting data packets by IoT device due to the difference of acknowledgement mechanisms of different platforms, this paper establishes an equivalent model, that is, in different environments, in order to ensure the success of one-time transmission of a data packet, IoT devices achieve equivalent realization by increasing the transmission power instead of retransmission.

According to the different snr, the NB-IoT network sends data at two rates, corresponding to BPSK modulation (Binary Phase Shift Keying) and QPSK modulation (Quadrature Phase Shift Keying). When the channel quality is relatively poor, NB-IoT network uses BPSK modulation, and the minimum transmission power consumption can be obtained when the BER (bit error ratio) requirement is met in different channel states [17].

$$P_{ber}(s_i) \le 0.5 \cdot erfc\left(\sqrt{\frac{snr_i \cdot P_i'}{\sigma}} \right) \tag{9}$$

Among them, $P_{ber}(s_i) = \frac{ber_i}{\sigma}$ is the probability of error rate, and $erfc(x) = \frac{2}{\sqrt{\pi}} \cdot \int_0^x e^{-\eta^2} d\eta$ is a Gaussian error function. ber is the allowable error rate of the wireless channel, and σ is the interference noise power. P_i' 'is the equivalent transmission power to ensure that data packet is not lost and not retransmitted while the bit error rate is met. When the channel quality is relatively good, NB-IoT network uses QPSK modulation to obtain the minimum transmission power consumption when meeting the BER requirement in different channel states, namely:

$$P_{ber}(s_i) \le 0.2 \cdot exp\left(\frac{-1.6 \cdot snr_i \cdot P_i'}{3 \cdot \sigma} \right) \tag{10}$$

When the device delays sending the *i*-th data packet, the power consumption of the data packet at this time is:

$$W_i = I_0 \cdot \Delta\tau_i + \frac{P_i'}{v} \cdot T_f \tag{11}$$

where *I0* is the working current of the device in idle state $\Delta\tau_i$, and when the device is in $\Delta\tau_i$, the idle current of the device in $\Delta\tau_i$ time is I_0. v is the device voltage, and T_f is the time for sending a data packet when the device is connected.

When the device immediately sends the ith data packet, the power consumption of the data packet at this time is:

$$W_i = I_0 \cdot \Delta\tau_i + \frac{P_i'}{v} \cdot T_f, \Delta\tau_i = 0 \tag{12}$$

After sending all data packets of all samples, the total power consumption in one day can be uniformly expressed as:

$$W_{day} = \sum_{m}^{M} \sum_{i=1}^{n} \left(\Delta\tau_i \cdot I_0 + \frac{P_i'}{v} \cdot T_f \right) \tag{13}$$

where M is the number of samples sent in one day, and n is the number of data packets corresponding to one sample. The immediate reward function of t_i is defined as the negative value of power consumption of sending a single data packet, that is, $R_i = -W_i$. The reward function of daily data transmission is the negative value of total power consumption, that is, $R_{day} = -W_{day}$. The goal of the system is to find the most effective policy for sending data packet sequence, so as to obtain the maximum benefit, that is, the minimum total power consumption.

4.6 Device Location Policy

If the expected life cycle of the IoT device is d days and the battery capacity is W_{total}, then $d = W_{total}/W_{day}$, that is, $D = d/365$ years. If the IoT device only sends one sample every day, the location decision can be obtained by the following formula:

$$W_{total} = \sum_{j=0}^{d} \sum_{i=1}^{n} \left(\Delta\tau_i \cdot I_0 + \frac{P_i'}{v} \cdot \tau_0 \right) \tag{14}$$

Because the battery capacity W_total of the device is known, the optimal $\Delta\tau_i$ and P_i' scheduling can be obtained by adopting an optimal policy when the expected number of life cycle of the IoT device is given.

Under the condition that the battery capacity, expected service life cycle and data amount of the device are all determined, whether IoT device can be installed in a certain location depends entirely on whether $\Delta\tau_i$ and P_i' have a feasible solution. In order to delay the service life of device, it is necessary to seek the optimal solution, that is, the optimal policy.

5 OPTIMAL DECISION-MAKING BASED ON FEW-SHOT REINFORCEMENT LEARNING

According to Eq. (13), the power consumption of IoT device is related to the number of samples. To solve the MDP problem, a large number of samples is needed. However, in order to reduce power consumption, IoT device can only send a small amount of data.

In order to solve this contradiction, this paper adopts PPO2 (Proximal Policy Optimization) algorithm [18] based on prior distribution to solve the last MDP problem, so as to achieve the optimal decision through few-shot learning. Because of the causal a priori or structural relationship, when there is a new scene that has not happened, it is not to learn from scratch, but to have a higher learning starting point.

In the process of random and aligned MDP, because of the randomness of state s and action a, the policy can be expressed by conditional distribution $\pi(a|s)$, where π represents the mapping from states to action a. Next, define a kind of neural network to fit the policy.

5.1 Sequence Expression of Seq2seq Neural Network

IoT device usually divides the data to be sent into a sequence of n data packets, and then it is necessary to make sending decisions for these n data packets in turn to form an action sequence. In this paper, seq2seq (sequence to sequence) neural network is used to express the sequence of the above process, that is, a sequence is input and a sequence is output. The network structure is shown as follows [19].

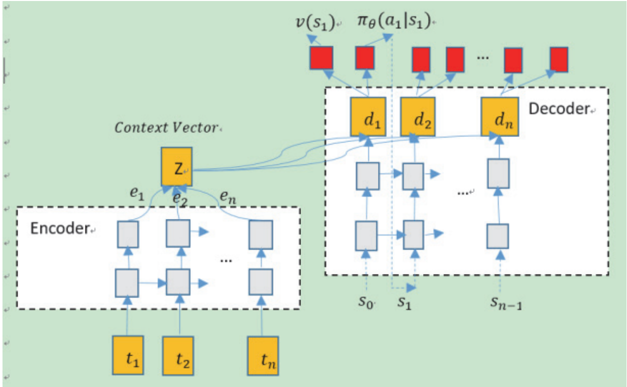

Figure 1 Architecture of the seq2seq neural network

Seq2seq neural network consists of encoder and decoder, both of which are RNN network (Recurrent Neural Network). Let the parameter of the neural network be θ, then the conditional probability of outputting the best action a can rewrite $\pi_\theta(a|s)$ when the state s is input. According to the input t_i, the neural network first learns and memorizes by the encoder, then outputs d by the decoder according to the memory of the network, and then respectively passes through two different activation functions, corresponding to the output state value function $v_\pi(s)$ and the action sequence probability $\pi_\theta(a|s)$.

In order to reflect the characteristics of the data to be sent, the data packet sequence can be converted into an embedded vector sequence and input into the neural network. The input t_i = [data packet number i, data packet size] is a two-dimensional vector.

For different types of devices, the amount of data to be sent may be different, and the corresponding packet sizes are different. The size of the last data packet sent by a device may be different from that of the first $n - 1$ data packets, which is the margin of the amount of data to be sent divided by $n-1$.

The encoder and decoder [20] are denoted as f_{enc} and f_{dec} respectively, then, the output of the encoder is:

$$e_i = f_{enc}\left(t_i, e_{i-1}\right) \tag{15}$$

The output of the decoder is:

$$d_j = f_{dec}\left(z_j, s_j, a_{j-1}, d_{j-1}\right) \tag{16}$$

Considering the relevance of successive environments, the input of the decoder consists of four parts, including the weighted sum z_j of the encoder output, the current environment s_j, and the decision execution results $d_{(j-1)}$ and $a_{(j-1)}$ of the previous step. z_j is the context of the j-step decoder, which contains the attributes of the data to be sent. Since the initial state s_0 affects the output d_1 of the decoder, and then the whole sequence d_j, a_j and $v_\pi(s_i)$, it is necessary to take the current state as one of the input variables of the decoder to optimize the decoding accuracy.

The output of seq2seq neural network is an n-dimensional vector d. After nonlinear activation of this vector, the n-dimensional probability vector π_θ and the sum function $v_\pi(s_i)$, are obtained respectively, in which the probability vector π_θ corresponds to the probabilities of different a for decision actions, and the sum of probabilities is 1. Then, the decision action $a_j = \text{argmax}_{a}(\pi_\theta)$ in step j can be obtained by greedy algorithm.

5.2 Parameter Update and Optimal Policy of Few-shot Learning

PPO2 algorithm defines the objective function of MDP as:

$$J = E\left(R_{day}\right) = E\left(-W_{day}\right) \tag{17}$$

In which the daily reward is defined as:

$$R_{day} = \sum_{m=1}^{M}\sum_{i}^{n} W_m(i) \tag{18}$$

In which M is the number of samples, each sample is divided into a sequence of n data packets for transmission, m is the sample number, i is the data packet number, and $W_m(i)$ is the power consumption required for transmitting a certain data packet.

Because the IoT device can only send a small number of samples every day, the number of samples is limited, which cannot satisfy the law of large numbers. In order to save power consumption, this paper regularizes the objective function of MDP with prior distribution to reduce the required number of samples.

The layout and location of IoT device has the following prior distribution:

1) The initial value s_0 of the environment can be set near the mean value, because the installation environment of the IoT device is similar, so this parameter can be generalized.

2) All the power consumption generated by the device sending data is positive, and all the corresponding benefits are negative, which leads to insufficient punishment. Therefore, by increasing the benefit deviation corresponding to the mean value, the expected reward is adjusted to positive and negative distribution.

3) The prior knowledge of samples shows that the power consumption function has an environmental inflection point. When snr_i is lower than this inflection point, the power consumption increases steadily; when snr_i is higher than the inflection point, the power consumption increases rapidly, so the MDP random distribution has a segmented structure. In this paper, Softplus distribution is introduced to model structure regularization parameters to fit prior knowledge, and Rectifier distribution is its special case, and refers to Fig. 2.

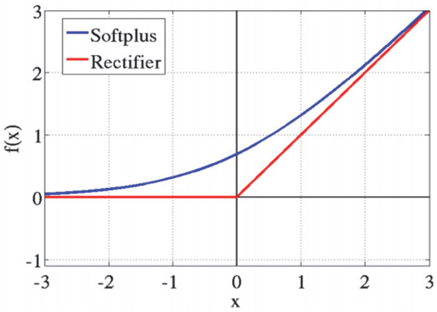

Figure 2 A priori distribution of environment

4) Maximum value constraint. When the signal quality of the device installation point is lower than a certain threshold, the power consumption is very large, so s_i needs the maximum limit.

Let each sample be divided into data packet sequence, $q = (q_1, q_2, \ldots, q_i)$, and the objective function of the m-th sample can be defined as:

$$J\left(\theta_m\right) = E_q\left(R_{\theta_m} - b_{\theta_m}\right) - \beta \cdot KL_q\left(\theta_m\right) \tag{19}$$

where $b_{(\theta m)}$ is the reward corresponding to the mean value, which is used to transform the positive distribution reward into the positive and negative distribution. Reward $R_{(\theta m)}$ [21] is expressed as:

$$R_{\theta_m} = \sum_{i=1}^{n} min\left(\omega_i \cdot A_i, clip_{1-\varepsilon}^{1+\varepsilon}\left(\omega_i \cdot A_i\right)\right) \tag{20}$$

Here, ω_i is the probability ratio between target policy and sample policy, and it is the adjustment coefficient of variance and deviation.

$$\omega_i = \frac{\pi_{\theta_m}\left(a_i \middle| s_i\right)}{\pi_{\theta_m^0}\left(a_i \middle| s_i\right)} \tag{21}$$

Advantage function A_i uses GAE estimation.

$$A_i = \sum_{j=0}^{n-i+1}\left(\gamma \cdot \omega_i\right)^j \cdot \left(r_{i+j} + \gamma \cdot v_\pi\left(s_{i+j+1}\right) - v_\pi\left(s_{i+j}\right)\right) \tag{22}$$

where γ is the discount coefficient of advantage function, r is the immediate reward. According to Eq. (13), the immediate reward function $r_{(i+j)}$ is:

$$r_{i+j} = -W_m\left(i+j\right) \tag{23}$$

KL function is the regularization constraint in the objective function, which is defined as:

$$KL\left(\theta_m\right) = E_q\left[\sum_{i=1}^{n}\left(v_\pi\left(s_i\right) - \bar{v}_\pi\left(s_i\right)^2\right)\right] \tag{24}$$

Among them:

$$\bar{v}_\pi\left(s_i\right) = \sum_{j=1}^{n-j+1} \gamma^j \cdot r_{i+j} \tag{25}$$

based on prior Softplus distribution, normalized β is defined as:

$$\beta = \log\left(1 + \mathrm{e}^{snr_i}\right) \Big/ snr_{N-1} \tag{26}$$

For each sample, the gradient update formula for few-shot learning can be given as follows:

$$\theta_m{}' = \theta_m + \alpha \cdot \nabla_{\theta_m}\left(J\left(\theta_m\right)\right) \tag{27}$$

where α is the learning rate, that is, the gradient descent factor. After the parameters of seq2seq network are iterated and converged, the output of the network is the action under the optimal policy.

The flow of few-shot reinforcement learning algorithm based on prior distribution is as follows.

The task distribution of a given sample is $\rho(T)$.

The parameter θ of the neural network is randomly initialized, and the initial state s_0 and the corresponding power consumption value W_0 are given.

Collect Mdata samples $\{T_0, T_1, \ldots, T_M\}$from $\rho(T)$.

For each data sampleT_m, do.

Let the sampling policy $\pi_{\theta m0} = \theta$, and use the sampling policy to collect data samples, and the data packet sequence corresponding to a sample is $q = (q_1, q_2, \ldots, q_i)$.

Use Eq. (27) based on prior distribution to calculate the policy of seq2seq network and update parameters θ_m.

Perform gradient iteration on q to learn a better policy, $\pi_{\theta m}$

If $snr_i > snr_{(N-1)}$, re-select the action a_i, otherwise $\theta = \pi_{\theta m}$.

Based on the prior structure distribution, the neural network can converge after only a few iterations, and learn the better policy $\pi_{\theta m'}$

end for

In the above algorithm flow, sample policy and learning policy are iterated interactively, and under the framework constraint of prior distribution, fast convergence can be achieved by using few-shot.

6 EXPERIMENT AND RESULT ANALYSIS

The list of simulation parameters used in this paper is as follows, including MDP model parameters and PPO2 algorithm parameters. During the simulation, a sample is divided into 10, 12, 15 and 20 data packet sequences for training in turn. In order to reduce the channel correlation, the minimum delay time segment t δ of data packet transmission is taken as 5 s.

Table 1 Simulation parameters

Parameter	Description value
Number of data packets	$n \in \{10, 12, 15, 20\}$
Minimum delay time segment	$\delta = 5$ s
Data packet transmission duration	$T_f = 1$ s
Packet size	< 200字节
SINR threshold	snr = [1.28 3.28 5.28 6.28]
Device voltage	$V = 3$
Device idle state current	$I_0 = 250$ μA
Sample number	$M = 5$
Expected life cycle	$d = 10 \times 365$ days
Learning rate	$\alpha \in$ [0.002, 0.005]
Reward discount factor	$\gamma = 0.9$
Advantage function discount coefficient	$\varphi = 0.95$
Clipping constant	$\varepsilon = 0.2$
Number of neurons	units = 256
Overfitting factor	0.5
Hidden units	Layers = 256
Encoding layer	Layer1 = 2
Decoding layer	Layer2 = 2

In the experiment, it is assumed that the environmental state s_max is −117 dbm, the average value of the signal coverage quality, that is, the initial state s_0 is −107 dbm, the battery capacity of the IoT device is 5000 mAh, and the battery voltage is 3.6 V.

The device delay state and channel state is encoded into an embedding, which is then input into the neural network. Fig. 3 shows the change process of the delay state of the device. When the action decision is to send immediately, the delay state of the device basically remains unchanged, because the sending time is very short; when the action decision is to delay sending, the delay state of the device has an obvious increase.

Channel is the environment of IoT device, and Fig. 4 shows the change process of channel state. When the channel quality is good, the SINR is relatively stable, and when the channel quality is poor, the SINR is more likely to fluctuate randomly.

The power required for data packet transmission is related to the state and environment. Fig. 5 shows the power values required for data packet sequence transmission when the environment changes randomly, and the equivalent model when the transmission is unsuccessful and retransmission is needed.

When the data packet is delayed, the power consumption is related to the current in idle state and waiting time. When sending data packets immediately, the required power consumption is related to sending power, device voltage and sending time.

Fig. 6 shows the corresponding immediate reward when sending the data packet sequence, and maximum reward is equivalent to minimum power consumption.

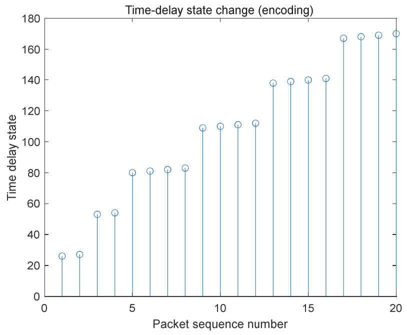

Figure 3 Time-delay stat change process of data packet sequence

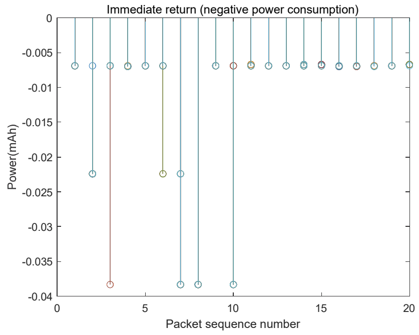

Figure 4 Channel state change process of data packet sequence

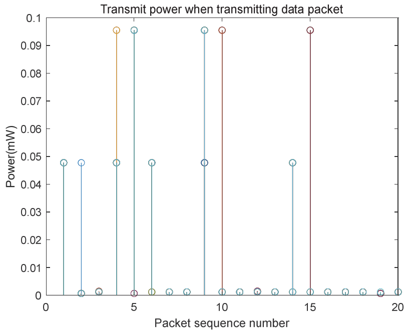

Figure 5 Transmit power of data packet sequence

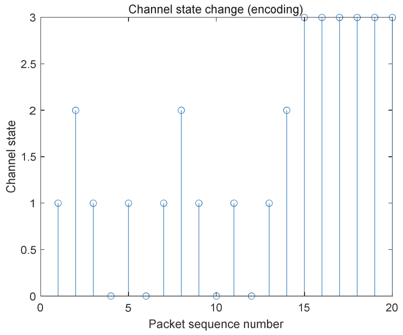

Figure 6 Immediate reward of packet sequence decision

After the MDP model is established, it is solved by PPO2 algorithm based on prior distribution. The parameters of neural network converge with the increase of iteration times, and the policy is gradually optimized.$b_{\theta m}$ is the expected reward corresponding to the mean value and $b_{\theta m} = 6$ in the experiment. The difference between the training target and $b_{\theta m}$ can be defined as a loss function.

Fig. 6 is the convergence process of neural network parameters based on normal random distribution of state, with a learning rate of 0.002, in which a sample is divided into 10, 12, 15 and 20 data packet sequences respectively. When the loss function tends to be constant, the parameters of the neural network converge to the optimal value, and the expression of the neural network corresponds to the optimal policy.

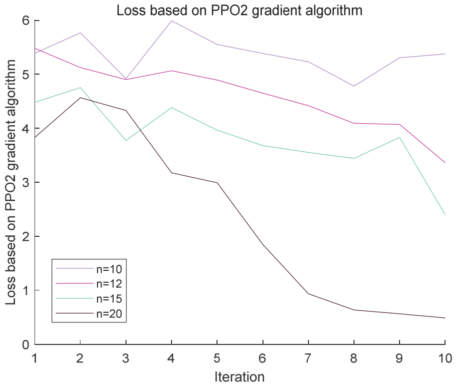

Figure 7 Convergence process of normal distribution without prior

Next, Fig. 8 shows the convergence process of the loss function after adjusting the state to the prior distribution, at which the learning rate of few-shot is 0.002. This is because the dynamic range decreases and the number of samples required when the loss function tends to be stable decreases. Fig. 9 is an iterative process when the learning rate of few-shot is 0.005, and the convergence is faster when the learning rate increases.

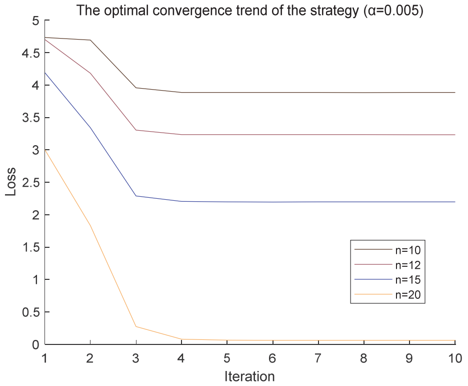

Figure 8 Convergence process of small sample learning (learn rate 0.002)

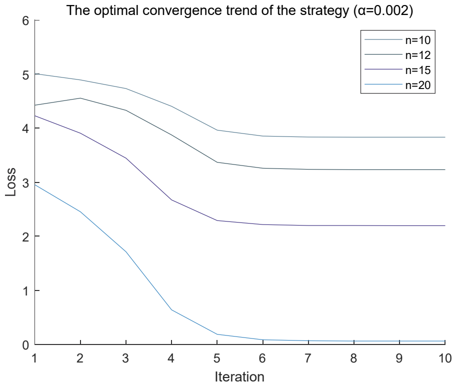

Figure 9 Convergence process of small sample learning (learn rate 0.005)

Fig. 8 and Fig. 9 show that the more data packets, the easier it is for neural network parameters to converge. This is because NB-IoT network has the highest transmission efficiency when a single data packet size is less than 200 bytes. On the one hand, when the amount of data to be transmitted is fixed, the more data is split, that is, the smaller the data packet, the smaller the power cost of transmission. On the other hand, when the amount of data to be transmitted is large, NB-IoT network will split the data into more data packets, which is more suitable for the requirements of NB-IoT network.

Take Smart Water as an example, the IoT device sends a fixed amount of data of one sample every day, and divides the sample into 20 data packets. From the above simulation results, it can be seen that when the sensing model and algorithm in this paper are not used, the power consumption required by the terminal to send a data packet is 0.04 mAh, referring to Fig. 5. Based on the life cycle of the terminal of 10 years, the required battery capacity is $0.04 \times 20 \times 365 \times 10 = 2920$ mAh. In this way, when the battery capacity is 5000 mAh, the device can send 1.71 samples every day. When the perception model and algorithm in this paper are adopted, it can be seen from Fig. 5 that the average power consumption required by the terminal to send a data packet is 0.018 mAh, which can support the device to send 3.7 samples every day.

Based on the reinforcement learning model and algorithm in this paper, when the amount of data to be sent is constant, the average power consumption required by the device to send a data packet is less. Therefore, under the same expected life cycle of the device, the signal coverage condition at the installation location of the device can be allowed to be worse, that is, the installation threshold can be lowered to support weaker signal coverage areas where IoT device or communication module can be installed.

7 CONCLUSION

In this paper, starting from the practical problem of intelligent water device layout and location, the established MDP model is solved by PPO2 algorithm with prior distribution. Firstly, the paper establishes a system model based on MDP, that is, Markov sequence decision model, and gives the definitions of the state, action and reward function of the model. Then, using seq2seq neural network to express the policy of the sequence decision, and based on the input sequence and its embedding, the decision-making problem of the output sequence is studied with few-shot reinforcement learning. In the experiment part, the solution process and results of PPO2 algorithm based on prior distribution are given, and it is concluded that when the life cycle and battery capacity of the IoT device are known, the optimal solution of the layout and location of the IoT device can be obtained. The above models and methods can be applied to practical management projects and problems in various narrowband Internet of Things scenarios.

8 REFERENCES

[1] Mothku, S. K. & Rout, R. R. (2019). Markov decision process and network coding for reliable data transmission in wireless sensor and actor networks. *Pervasive and Mobile Computing*, *56*, 29-44. https://doi.org/10.1016/j.pmcj.2019.03.003

[2] Hassan, M. B., Ali, E. S., Mokhtar, R. A., Saeed, R. A., & Chaudhari, B. S. (2020). Nb-iot: concepts, applications, and deployment challenges. *In LPWAN Technologies for IoT and M2M Applications*, 119-144. https://doi.org/10.1016/B978-0-12-818880-4.00006-5

[3] China Metrology association water meter industry group standard, Technical Guide for Layout and Location, Acceptance and Use of NB-IoT Automatic Meter Reading System, TCMA SB 040-2019

[4] Yasin, H. M., Zeebaree, S. R., Sadeeq, M. A., Ameen, S. Y., Ibrahim, I. M., Zebari, R. R., & Sallow, A. B. (2021). IoT and ICT based smart water management, monitoring and controlling system: A review. *Asian Journal of Research in Computer Science*, *8*(2), 42-56. https://doi.org/10.9734/AJRCOS/2021/v8i230198

[5] Jan, F., Min-Allah, N., & Düştegör, D. (2021). Iot based smart water quality monitoring: Recent techniques, trends and challenges for domestic applications. *Water*, *13*(13), 1729. https://doi.org/10.3390/w13131729

[6] Singh, M. & Ahmed, S. (2021). IoT based smart water management systems: A systematic review. *Materials Today: Proceedings*, *46*, 5211-5218. https://doi.org/10.1016/j.matpr.2020.08.588

[7] Benyezza, H., Bouhedda, M., &Rebouh, S. (2021). Zoning irrigation smart system based on fuzzy control technology and IoT for water and energy saving. *Journal of Cleaner production*, *302*, 127001. https://doi.org/10.1016/j.jclepro.2021.127001

[8] Alshehri, M., Bhardwaj, A., Kumar, M., Mishra, S., & Gyani, J. (2021). Cloud and IoT based smart architecture for desalination water treatment. *Environmental Research*, *195*, 110812. https://doi.org/10.1016/j.envres.2021.110812

[9] Pincheira, M., Vecchio, M., Giaffreda, R., &Kanhere, S. S. (2021). Cost-effective IoT devices as trustworthy data sources for a blockchain-based water management system in precision agriculture. *Computers and Electronics in Agriculture*, *180*, 105889. https://doi.org/10.1016/j.compag.2020.105889

[10] Akhter, F., Siddiquei, H. R., Alahi, M. E. E., Jayasundera, K., & Mukhopadhyay, S. C. (2021). An IoT-enabled portable water quality monitoring system with MWCNT/PDMS multifunctional sensor for agricultural applications. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2021.3069894

[11] Akhter, F., Siddiquei, H. R., Alahi, M. E. E., & Mukhopadhyay, S. C. (2021). Design and development of an IoT-enabled portable phosphate detection system in water for smart agriculture. *Sensors and Actuators A: Physical*, *330*, 112861. https://doi.org/10.1016/j.sna.2021.112861

[12] Priyanka, E. B., Thangavel, S., Madhuvishal, V., Tharun, S., Raagul, K. V., & Krishnan, S. (2021). Application of integrated IoT framework to water pipeline transportation system in smart cities. *In Intelligence in Big Data Technologies-Beyond the Hype*, 571-579. https://doi.org/10.1007/978-981-15-5285-4_57

[13] Watanabe, A. O., Ali, M., Sayeed, S. Y. B., Tummala, R. R., & Pulugurtha, M. R. (2020). A review of 5G front-end systems package integration. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, *11*(1), 118-133. https://doi.org/10.1109/TCPMT.2020.3041412

[14] Liu, Y., Ding, J., & Liu, X. (2020, October). A constrained reinforcement learning based approach for network slicing. *In 2020 IEEE 28th International Conference on Network Protocols (ICNP), IEEE*, 1-6 https://doi.org/10.1109/ICNP49622.2020.9259378

[15] Jiang, Z. H. U., Tingting, W. A. N. G., Yonghui, S. O. N. G., & Yali, L. I. U. Transmission scheduling scheme based on deep Q learning in wireless network. *Journal on Communications*, *39*(4), 35.

[16] Chung, S. T. & Goldsmith, A. J. (2001). Degrees of freedom in adaptive modulation: a unified view. *IEEE Transactions on Communications*, *49*(9), 1561-1571. https://doi.org/10.1109/26.950343

[17] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. https://doi.org/10.48550/arXiv.1707.06347

[18] Chen, C., Zhang, Y., Wang, Z., Wan, S., & Pei, Q. (2021). Distributed computation offloading method based on deep reinforcement learning in ICV. *Applied Soft Computing*, *103*, 107108. https://doi.org/10.1016/j.asoc.2021.107108

[19] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473. https://doi.org/10.48550/arXiv.1409.0473

[20] Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438. https://doi.org/10.48550/arXiv.1506.02438

[21] Wang, J., Hu, J., Min, G., Zomaya, A. Y., & Georgalas, N. (2020). Fast adaptive task offloading in edge computing based on meta reinforcement learning. *IEEE Transactions on Parallel and Distributed Systems*, *32*(1), 242-253. https://doi.org/10.1109/TPDS.2020.3014896

Contact information:

Shikai XING, PhD candidate
Beijing Jiaotong University,
No.3 Shangyuancun Haidian District Beijing 100044 P. R. China
E-mail: xingshikai@163.com