A Deep Reinforcement Learning based Adaptive Transmission Strategy in Space-Air-Ground Integrated Networks

Mengjie Liu, Member, IEEE, Gang Feng, Senior Member, IEEE, Lei Cheng, Shuang Qin, Member, IEEE,

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838681

***Abstract*—Space-air-ground integrated network (SAGIN) is an emerging architecture for future wireless communication systems, by exploiting the advantages of combined satellite, aerial and terrestrial communications. In such an integrated system, there may exist intra-cell, inter-cell and inter-system interferences, leading to unsatisfactory system performance. On the other hand, it is very challenging to optimize the system performance due to the unique characteristics of SAGINs, such as time-varying links, heterogeneous resources, and three-dimensional network architecture. In this paper, we propose a deep reinforcement learning based intelligent adaptive transmission strategy. We first formulate the adaptive transmission strategy problem (ATSP) with the aim to maximize the system throughput while meeting the delay and reliability requirements of packets. The re-parameterization method based deep deterministic policy gradient (RPDDPG) algorithm is proposed for achieving better performance compared with the relaxation-based DDPG algorithm. Numerical results demonstrate the performance improvement of the RPDDPG algorithm compared with the conventional relaxation-based DDPG algorithm and a heuristic algorithm.**

***Index Terms*—Deep reinforcement learning, space-air-ground integrated networks, adaptive transmission.**

I. Introduction

To provide anytime and anywhere communication experience for mobile users, one of the goals of 6G is to achieve a seamless global coverage. Recently, the space-air-ground integrated network (SAGIN) has been widely acknowledged as an effective solution for full coverage, mobile device management and service disruption in harsh environment [1]. Nevertheless, the system performance of the integrated network may become unsatisfactory due to serious interference, including intra-cell, inter-cell and inter-system interference [2], [3].

In such SAGINs, adaptive transmission has been proposed as an effective technique to improve service quality, where there exist two transmission modes [4]. In the direct transmission mode, a user can directly access to a satellite. In the cooperative transmission mode, a user can access to an aerial components (AC) as a relay or both of a satellite and an AC to enhance the connectivity. However, due to heterogeneous resources, time-varying links, dynamic environment and three-dimensional network architecture of SAGINs, it is very challenging to design an efficient adaptive transmission strategy. Moreover, since the throughput performance is affected by highly coupled transmission mode selection, spectrum and power resource allocation, a joint optimization problem should be investigated to make optimal decisions.

Recently, there have been some initial research efforts to address this challenge. Especially, machine learning (ML) has been emerging as a promising paradigm to address many important issues which cannot be readily solved by using traditional static optimization techniques. Among ML techniques, deep reinforcement learning can be exploited to perform multi-dimensional resource management in dynamic environments, which reduces the cost of human involvement [2]. For solving the adaptive transmission problem in SAGINs which is rather dynamic yet complex, ML is expected to be fairly effective and efficient. When designing the deep reinforcement learning based strategies, a challenge is to identify or construct an appropriate deep reinforcement learning architecture for SAGINs, which is more complex compared with terrestrial networks. Recently, deep deterministic policy gradient (DDPG) based schemes have been identified as an efficient deep reinforcement learning architecture in wireless networks [5] [6]. In statistics, discrete variables are used to define non-distinguishable variables, such as access node selection. However, existing deep reinforcement leaning methods are assisted by a relaxation-based method to handle discrete variables, which may reduce the convergence speed and cause deviation from the optimal solution [6]. This observation inspires us to develop improved assisted methods for DDPG to achieve near-optimal solution efficiently.

In this paper, we develop a re-parameterization method based deep deterministic policy gradient (RPDDPG) algorithm for the adaptive transmission strategy for SAGINs. We first formulate the adaptive transmission strategy problem (ATSP) as a mixed-integer stochastic optimization problem with the aim to maximize the system throughput while meeting the delay and reliability requirements of traffic flows. The proposed RPDDPG algorithm takes the advantage of *probability distribution* of relay selection and *re-*

The authors are with the University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou 313001, P. R. China. This work was supported by the National Science Foundation of China under Grant number 62071091, the Huawei Project TC2021031600, the Research Funds for the Central Universities under Grant number ZYGX2020ZB044.

parameterization method. The RPDDPG has two significant features. First, the RPDDPG replaces discrete variables with continuous variables which have probabilistic meanings. Second, the final output values of integer variables are generated according to the learned probability distribution of the continuous variables rather than the conventional discretization of output values of continuous variables. With the two features, the policy learned by the RPDDPG algorithm can efficiently perform the better transmission modes as well as the resource allocation for traffic flows, than using the coarse relaxation-based DDPG. Numerical results demonstrate the effectiveness of the RPDDPG compared with the conventional relaxation based DDPG algorithm.

II. Model and Problem Formulation

As illustrated in Fig.1, we consider an SAGIN model consisting of low earth orbit (LEO) satellites, aerial components (ACs), terrestrial base stations (BSs) and user equipments (UEs) [7]. We assume that at any specific time only one satellite participates in adaptive transmission. In this paper, we focus on the down-link transmission from one LEO satellite to UEs. Two transmission modes as described in the introduction section are considered. In this study, time is divided into slots with the same length and a certain number of slots form an episode. The decision is made by using the deep reinforcement algorithm at each time slot.

In the following, we first specify the channel, delay and outage probability models respectively and then we formulate the adaptive transmission strategy problem.

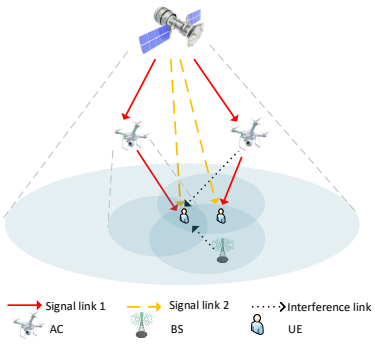

Fig. 1. Adaptive transmission in the SAGIN system

A. Channel Model

In SAGINs, three segments, i.e., space, aerial and terrestrial communications could be involved, and thus there are three corresponding types of channel model to be considered.

Space channel: For characterizing space channel, shadowed rice fading channel is adopted to model the signal propagation from satellite to a UE [8]. Meanwhile, the space-ground link is modeled as a shadowed rice fading channel [8]. Denote h_{su} as the channel gain at link satellite $\rightarrow$ UE. The probability distribution function (PDF) of $|h_{\mathrm{su}}|^2$ can be found in [8]. When a UE receives signal from the satellite, it may suffer from the co-channel interference (CCI) generated from neighboring BSs and ACs. The received signal to interference plus noise ratio (SINR) at a UE from satellite can thus be expressed as

$$\gamma_{\mathrm{su}} = \frac{P_{\mathrm{su}} |h_{\mathrm{su}}|^2}{I_{\mathrm{a}} + I_{\mathrm{b}} + N_0}, \tag{1}$$

where P_{su} is the transmit power from satellite to UE; N_0 is the average white Gaussian noise power; I_{a} and I_{b} are the interference from ACs and BSs, respectively; we have $I_{\mathrm{a}} = \frac{P_{\mathrm{a}}}{Q_{au}} |h_{\mathrm{au}}|^2$ and $I_{\mathrm{b}} = \sum_{i=1}^{N_b} P_{\mathrm{i}} r_i^{-\alpha} |h_{iu}|^2$, where P_a is the total transmit power of interfering ACs, P_{i} is the transmit power at interfering BS i, r_i is the distance between BS_i and UE, the PDF of $|h_{iu}|^2$ can be given by Generalized-K distribution, α is the distance factor, and Q_{au} is the absolute power loss at link AC $\rightarrow$ UE. The channel model of link satellite $\rightarrow$ AC is highly similar to the channel model of link satellite $\rightarrow$ UE [9]. Denote h_{sa} as the channel gain at link satellite $\rightarrow$ AC. Thus, the PDF of $|h_{\mathrm{sa}}|^2$ can also be found in [8].

Aerial channel: For simplicity, we assume that a UE under the overlapping area of ACs accesses to the nearest AC. Since the movement of AC is slow when serving UE, the Doppler frequency shift is assumed to have a negligible impact on channel [7]. When a UE receives signal form AC, the CCI is generated from neighboring BSs and the satellite. Hence, the received SINR at the UE from the AC can thus be expressed as

$$\gamma_{\mathrm{au}} = \frac{P_{\mathrm{au}} Q_{au}^{-1} |h_{\mathrm{au}}|^2}{I_{\mathrm{a}} + I_{\mathrm{b}} + N_0}, \tag{2}$$

where I_{a} and I_{b} denote the interference from ACs and BSs, respectively; h_{au} is the channel gain at link AC $\rightarrow$ UE. The PDF of h_{au} can be found in [10].

Terrestrial channel: For the channel between UE and BS, we adopt the classic Rayleigh fading and Gamma shadowing channel model. According to [11], the aggregate interference (i.e., I_b) from Poisson point process-based BSs can be approximated as the Gamma distribution with shape factor $k_b = \frac{(E[I_b])^2}{Var[I_b]}$ and scale parameter $\eta_b = \frac{(Var[I_b])^2}{E[I_b]}$, *i.e.*, $I_b \sim \Gamma(k_b, \eta_b)$, which can be given by $f_{I_b}(x) = \frac{x^{k_b-1} e^{-\frac{x}{\eta_b}}}{\Gamma(k_b) \eta_b^{k_b}}$.

B. Outage Probability Model

In adaptive transmission, an outage occurs when the UE can neither receive signal from the relay AC nor from the satellite. The outage probability is defined as the probability that the received SINR at the UE falls below a certain threshold γ_{th}, which is given by $P_{out}(\gamma_{th}) = \Pr\{\gamma \leq \gamma_{th}\} = F_r(\gamma_{th})$. Hence, the expression of link outage probability can be given by

$$P_{out}(\gamma_{th}) = \begin{cases} \Pr\{\gamma_{su} \leq \gamma_{th}\}, & \text{if} \quad \gamma_{sa} < \gamma_{th} \\ \Pr\{\gamma_{au} \leq \gamma_{th}\}, & \text{if} \quad \gamma_{su} \leq \gamma_{th}. \end{cases} \tag{3}$$

The cumulative distribution function (CDF) of (3) can be rewritten as

$$F_\gamma(\gamma_{th}) = F_{\gamma_{su}}(\gamma_{th})(F_{\gamma_{sa}}(\gamma_{th}) + F_{\gamma_{au}}(\gamma_{th})). \tag{4}$$

Due to the page limit, the derivation of outage probability under the above two transmission modes is omitted.

C. Delay Model

Let $\mathcal{U}(t) = \{1, 2, \cdot, u, \cdot, U(t)\}$ be the set of UEs within the coverage of the satellite at time slot t, where $U(t) = |\mathcal{U}(t)|$. Let $\mathcal{A}(t) = \{1, 2, \cdot, a, \cdot, A(t)\}$ be the set of ACs within the coverage of the satellite at time slot t, where $A(t) = |\mathcal{A}(t)|$. Let $\mathcal{K}(t)$ be the set of generated packets at the satellite at time slot t. A packet k in SAGINs can be described by a tuple $p^k(t) = \{u^k(t), b^k(t), g^k(t), d^k(t)\}$, where $u^k(t)$ is the index of UE receiving packet k, $b^k(t)$ is the size of packet k, $g^k(t)$ is the maximum outage probability of the down link communication and $d^k(t)$ is the maximum delay of transmitting the packet, respectively.

For packet $k, k \in \mathcal{K}(t)$, let binary variables $m_{k,u}(t)$ and $m_{k,a}(t)$ be the satellite-UE and satellite-AC association modes respectively, where $m_{k,u}(t) = 1$ (or $m_{k,a}(t) = 1$) if the satellite associates with UE $u = u^k(t)$ directly (or associate AC a as a relay) at time slot t, and $m_{k,u}(t) = 0$ (or $m_{k,a}(t) = 0$) otherwise. Note that for packet k, if UE $u^k(t)$ is outside of the coverage area of any AC, we directly set $m_{k,a}(t) = 0$. When packet k is rejected by the system, we have $m_{k,u}(t) = m_{k,a}(t) = 0$. Let S be the available spectrum resources shared by the down link transmissions from the satellite to UEs $\mathcal{U}$ and ACs $\mathcal{A}(t)$. Denote $f^k_{s,u}(t)$ as the fraction of spectrum allocated to UE $u^k(t)$ by the satellite at time slot t. Namely, UE $u^k(t)$ can occupy spectrum resource $f^k_{s,u}(t)S$ to receive the packet k from the satellite at time slot t. Denote $f^k_{a,u}(t)$ as the fraction of spectrum allocated to UE $u^k(t)$ by the AC $a, a \in \mathcal{A}(t)$ at time slot t. Denote $f^k_{s,a}(t)$ as the fraction of spectrum allocated to AC a by the satellite at time slot t. In addition to spectrum allocation, proper amounts of power resources needs to be allocated by the satellite and ACs. Let $z^k_{s,a}(t)$ and $z^k_{s,u}(t)$ be the fraction of power resources allocated to packet k from the satellite to AC a and UE $u^k(t)$ at time slot t respectively.

We denote by $b^k(t)$ the size of packet k. Then, the transmission time of the packet at link $\xi \in \{su, au, sa\}$ can be expressed as

$$T^k_\xi(t) = b^k(t)/R_\xi(t), \tag{5}$$

where $R_\xi(t)(R_\xi(t) = Sf^k_\xi(t)e_\xi(t))$ is the corresponding down link transmission rate, $e_\xi(t)(e_\xi(t) = log_2(1 + \frac{P' z^k_\xi(t)|h_\xi(t)|^2}{I_a(t)+I_b+N_0}))$ is the spectrum efficiency at link ξ and P' is the transmit power of AC or satellite.

D. Problem Formulation

One of the major goals in the design of adaptive transmission in SAGINs is to maximize system throughput. We define that a packet is regarded to be successfully *completed* when it is successfully transmitted from satellite to the destined UE with satisfied service quality. Based on the above models, the optimal adaptive transmission strategy problem (ATSP) can be formulated as the following optimization problem (P) with the aim of maximizing the number of *completed* packets, while satisfying the resource capacity constraints and service quality.

$$P: \max_{\mathbf{m},\mathbf{f},\mathbf{z}} \sum_{k\in\mathcal{K}(t)} (1 - m_{k,u}(t)m_{k,a}(t))m_{k,u}(t)E(T^k_{su}(t), \tag{6}$$

$$F^t_{\gamma_{su}}(\gamma_{th})) + m_{k,u}(t)m_{k,a}(t)E(T^k_{coop}(t), P^t_{out}(\gamma_{th}))$$

$$s.t., \sum_{k\in\mathcal{K}(t)} m_{k,u}(t)f^k_{s,u}(t) + m_{k,a}(t)f^k_{s,a}(t) = 1, \tag{6a}$$

$$\sum_{k\in\mathcal{K}(t)} m_{k,a}(t)f^k_{a,u}(t) = 1, \tag{6b}$$

$$\sum_{k\in\mathcal{K}(t)} m_{k,u}(t)z^k_{s,u}(t) + m_{k,a}(t)z^k_{s,a}(t) = 1, \tag{6c}$$

$$\sum_{k\in\mathcal{K}(t)} m_{k,a}(t)z^k_{a,u}(t) = 1, \tag{6d}$$

$$m \in \{0, 1\}, \forall m \in \mathbf{m}, f, z \in [0, 1], \forall f \in \mathbf{f}, \forall z \in \mathbf{z}. \tag{6e}$$

where $E(x^k(t), y^k(t)) = H(x^k(t) - d^k(t))H(y^k(t) - g^k(t))$ with Heaviside step function $H(\cdot)$, indicates whether the QoS requirements of packets are satisfied; $F^t_{\gamma_{su}}(\gamma_{th})$ and $P^t_{out}(\gamma_{th})$ are the outage probability of link in the direct and cooperative transmission mode at time slot t respectively; $T^k_{coop}(t) = \max\{T^k_{su}(t), T^k_{sa}(t) + T^k_{au}(t)\}$; (6a)-(6d) are the resource capacity constraints and $\mathbf{m} = \{m_{k,u}(t), m_{k,a}(t)\}$, $\mathbf{f} = \{f^k_{s,u}(t), f^k_{s,a}(t), f^k_{a,u}(t)\}$ and $\mathbf{z} = \{z^k_{s,a}(t), z^k_{s,u}(t), z^k_{a,u}(t)\}$ are the sets of transmission mode selection, power and spectrum resource allocation matrices respectively.

III. Adaptive Transmission Strategy

The above ATSP is hard to be solved by using traditional methods, since i) the ATSP is a mixed-integer stochastic optimization problem; ii) the spectrum allocation is coupled with power resource allocation and iii) dynamic channel quality, mobility of ACs and QoS requirements of packets. Therefore, we resort to deep reinforcement learning based techniques. Specifically, we first re-model the ATSP as a Markov decision process (MDP). To deal with the action space consisting of both discrete and continuous integers, we design a re-parameterization method based DDPG (RPDDPG) algorithm which combines the DDPG algorithm with the re-parameterization method.

A. Problem Transformation

First, we transform the formulated ATSP into a Markov game, including a set of states **S** and a set of actions **A**. For a given state $s \in \mathcal{S}$, an optimal action is chosen from the action spaces by using policy $\pi : \mathbf{S} \to \mathbf{A}$.

1) *Environment State:* According to the ATSP, the environment state at time slot t, $s(t)$, can be given by

$$s(t) = \{\mathcal{U}(t), \mathcal{A}(t), p^1(t), \ldots, p^{|\mathcal{K}(t)|}(t)\}. \tag{7}$$

2) *Action:* The action at time slot t can be described as the follows. To guarantee the resource availability, we design the normalization module for action space in this section.

$$a(t) = \{m_{k,u}(t), m_{k,a}(t), f^k_{s,u}(t), f^k_{s,a}(t), f^k_{a,u}(t), z^k_{s,a}(t), z^k_{s,u}(t), z^k_{a,u}(t) | k \in \mathcal{K}(t)\}. \tag{8}$$

3) *Reward:* In practical scenarios, the learning process faces a problem that when the agent gets sparse rewards, it leads to slow or even ineffective learning. The step function is an example of a sparse reward function. An efficient way is shaping the reward function to get gradual feedback which helps it learn faster. Let $r : \mathbf{S} \times \mathbf{A} \rightarrow \mathbb{R}$ be the reward. We design the following reward according to the objective function.

$$r = \sum_{k \in \mathcal{K}(t)} (1 - m_{k,u}(t) m_{k,a}(t)) m_{k,u}(t) E'(T^k_{su}(t), P^k_{out,d}(t)) + m_{k,u}(t) m_{k,a}(t) E'(T^k_{coop}(t), P^k_{out}(t)) \tag{9}$$

where

$$E'(x^k(t), y^k(t)) = log_2(\frac{d^k(t)}{x^k(t)} + \delta) \cdot log_2(\frac{g^k(t)}{y^k(t)} + \delta). \tag{10}$$

$log_2(\frac{d^k(t)}{x^k(t)} + \delta)$ and $log_2(\frac{g^k(t)}{y^k(t)} + \delta)$ describe the satisfaction degree of the QoS requirement of packet k by action $a(t)$. If QoS requirement is satisfied, we have $\frac{d^k(t)}{x^k(t)} \geq 1$ and $\frac{g^k(t)}{y^k(t)} \geq 1$ under action $a(t)$ and thus $E'(x^k(t), y^k(t)) > 0$. Conversely, $E'(x^k(t), y^k(t)) < 0$. We adopt a logarithmic function in the reward function. The incremental rate of a logarithm element $E'(x^k(t), y^k(t))$ for packet k slows down when it reaches positive. In this case, the reward function will guide the system to allocate the resources to satisfy the QoS requirements of as many packets as possible. A small value δ is added to limit the minimum value of each reward element to $\log_2(\delta)$, which avoids the sharp fluctuation on the reward.

B. RPDDPG Solution

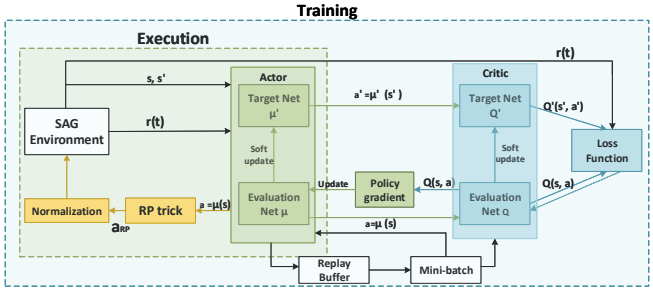

Fig. 2. The RPDDPG framework for solving the ATSP in the SAGIN

DDPG algorithm combines the advantages of policy gradient and deep Q-network, which can handle continuous and high-dimensional action space by using a gradient descent approach. Usually, integers in a formulated problem are relaxed to continuous variables when using the DDPG algorithm, and needs to be discretized as final outputs. For example, output 0.6 and output 0.9 are both rounded to 1, which means that different outputs lead to a same result. This may cause a deviation from the optimal solution. Intuitively, distinguishing different outputs (finer granularity of the relaxed variables) can improve the efficiency of the algorithm, as the error may come from the oversimplified rounding of the relaxed variables.

To address the problem, we design a re-parameterization method (RP) based DDPG algorithm. As illustrated in Fig.2, the RPDDPG framework is composed of three modules, *i.e.*, actor/critic, RP method and normalization module. The output action chosen by the actor is processed by the RP module and normalization module before calculating the reward. The details of the above modules are given as below. Please refer to [6] for the details of the Actor/Critic model. The RPDDPG algorithm is summarized in Algorithm 1.

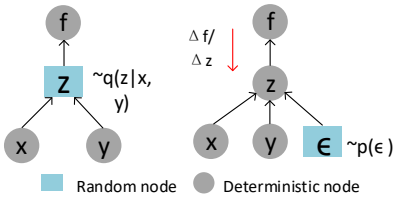

Fig. 3. Re-parameterization method in the neural network with back propagation

RP method: A straightforward way is to select discrete actions according to probability distribution, which is widely used in deep Q-learning methods. However, other problems arises: how to calculate the gradient in this way? How to update the network with back propagation (BP) if we can't calculate the gradient?

Intuitively, we first transform the integer variable in the action space into two new continuous variables. Specifically, $m_{k,u}(t)$ is transformed to $p^0_{k,u}$ and $p^1_{k,u}$ ($p^0_{k,u}, p^1_{k,u} \in [0, 1]$), which represent the probability of taking 0 and 1, respectively. Do the same for $m_{k,a}(t)$. Actually, we learn the distribution of discrete random variables $m_{k,u}(t)$ and $m_{k,a}(t)$. In this case, we get new action space with continuous variables:

$$a(t) = \{p^0_{k,u}(t), p^1_{k,u}(t), p^0_{k,a}(t), p^1_{k,a}(t), f^k_{s,u}(t), f^k_{s,a}(t), f^k_{a,u}(t), z^k_{s,a}(t), z^k_{s,u}(t), z^k_{a,u}(t) | k \in \mathcal{K}(t)\}. \tag{11}$$

Then, we move the step of selecting actions according to probabilities out of the neural network with BP by using the RP method. The specific operations of RP method are performed on the relaxed variables [12], $\{p^0_{k,u}, p^1_{k,u}, p^0_{k,a}, p^1_{k,a} | k \in \mathcal{K}(t)\}$:

1) Generate independent ϵ obeying a uniform distribution $U(0, 1)$.

2) Using Gumbel-Softmax method, generate noise $G = -\log(-\log(\epsilon))$ for each relaxed variable.
3) Add the noise to the each relaxed variables from the output, *i.e.*, $p' = p + G, p \in \{p^0_{k,u}, p^1_{k,u}, p^0_{k,a}, p^1_{k,a}\}$.
4) Assign probabilistic meaning to the relaxed variables by using softmax function:

$$\sigma_\tau(p^{0/1}_{k,u/a}) = \frac{e^{p^{0/1}_{k,u/a}/\tau}}{e^{p^0_{k,u/a}/\tau} + e^{p^1_{k,u/a}/\tau}}. \quad (12)$$

where τ is the temperature parameter.
5) Finally, generate the final output 0-1 variables according to the the above probability distribution. Get original action after RP method $a^{RP}_t = \{m, f, z\}$.

Normalization: To guarantee the total amount of resources allocated to the packets is no more than the maximal network capacity, we design normalization module. Based on m in the action a^{RP}_t, we do the normalization on the action's elements, *i.e.*, {f,z}, corresponding to the constraints (6a)-(6d) in the formulated problems.

Algorithm 1 RPDDPG Algorithm

Initialize critic network $Q(s,a)$ and actor network $\mu(s)$ weights θ^Q and θ^μ, replay buffer R, and transform the discrete variables to continuous variables.
repeat
  episode=episode+1.
  Initialize a random process $\mathcal{N}$ for exploration of action sets, and a uniform distribution $\mathcal{U}$ for exploration of action set.
  repeat
    $t = t + 1$.
    Select action $a_t = \mu(s_t)$ according to the current policy. Add noise to action and get new action $a_t = \{p, f, z\}$.
    Get $\sigma_\tau(p^{0/1}_{k,u/a})$ by using the operations of **re-parametrization method**. Select $m = \{m_{k,u}(t), m_{k,a}(t)\}$ according to $\sigma_\tau(p^{0/1}_{k,u/a})$. Get original action $a^{RP}_t = \{m, f, z\}$ after RP method.
    Do **normalization** for f and z.
    Execute action a^{RP}_t and get reward r_t and new state s_{t+1}. Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer R.
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_i) from R.
    Update critic by minimizing the loss function based on the minibath. Update the actor policy using the policy gradient based on the minibath. Update the target network
  until t=T
until episode=M

IV. Numerical Results and Analysis

In this section, we conduct simulation experiments to evaluate the proposed RPDDPG algorithm. In the first experiment, we compare the convergence performance of the RPDDPG algorithm with that of the DDPG algorithm in the training stage. Under the environment states, we train the RPDDPG model[1] based on Algorithm 1. Then, in the operation stage, we compare the throughput, outage probability and delay by using the policy learned by RPDDPG and DDPG. The learned policy of RPDDPG and DDPG algorithms are tested based on the new environment states with various resource capacity of the satellite/ACs. A heuristic scheme is adopted in the adaptive transmission strategy, where the probability of actions on each state is fixed based on large amounts of sampling data. We use a simulated SAGIN topology with 25 UEs, 10 ACs and 4 BSs. Referring to [10] [13], the parameters of the three kinds of channels are set. As a high learning rate speeds up the convergence of learning while it affects the convergence stability and a low learning rate may result in the convergence to a local optimal solution, we set all parameters in the TensorFlow according our simulation experience, which are listed in Fig.4.

Fig.5 and Fig.6 show the rewards and average outage probability of RPDDPG and DDPG algorithms as a function of episode in the training stage. As the replay buffer in the first episode is empty and the rewards are achieved based on the initial state, the rewards are small and fluctuate at the first 200 episodes of both RPDDPG and DDPG. As two DNNs' parameters are gradually approaching the optimal ones and the state-value function, the rewards become stable and high. We can see that the RPDDPG algorithm convergences more quickly than the DDPG algorithm. We also can see that the RPDDPG algorithm always achieves higher rewards and lower average outage probability than the DDPG algorithm. This validates that the RPDDPG algorithm learns two DNNs more effectively and efficiently than the DDPG algorithm.

Fig.7 shows the accumulated throughput as a function of episode in the operation stage. We can see that the accumulated throughput in a long term view achieved by the policy learned by RPDDPG is higher than that by DDPG. Fig.8 and Fig.9 show the average outage probability satisfaction ratios (OSR) and delay satisfaction ratios (DSR) achieved by RPDDPG, DDPG and random algorithms as a function of the amounts of power resources of the satellite, respectively. As shown in two figures, we can see that the OSR and DSR achieved by the policy learned by RPDDPG are higher than that by DDPG and are almost twice of that by the random one. Hence, in the operation stage, the policy learned by RPDDPG achieves higher throughput, OSR and DSR than that by DDPG, as validated in Fig.5 and Fig.6.

As shown in Fig.7-Fig.9, a higher satisfaction ratio can be achieved by the RPDDPG algorithm than the DDPG

[1]For the actor and the critic, we deploy two fully-connected hidden layers with [124,248] and [248,248] neurons, respectively. The output layers of actor and critic are activated by the *multiply* function and *dense* function.

TABLE I
PARAMETERS FOR THE LEARNING STAGE

Parameter	Value
Size of replay buffer	350
Size of mini-batch	80
Actor's learning rate	0.002 0.00001
critic's learning rate	0.001 0.00001
Reward discount factor	0.9

Fig. 4. Parameters for the learning stage

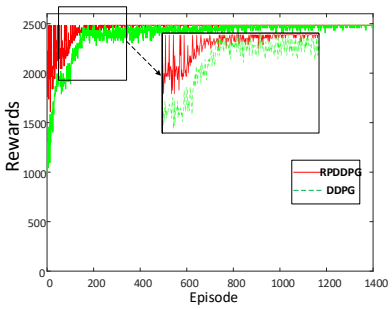

Fig. 5. Rewards vs episode at the learning stage

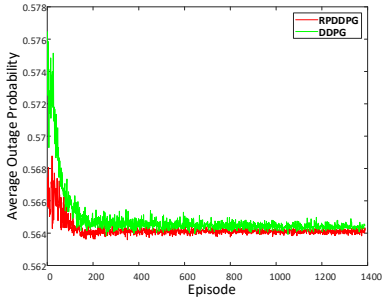

Fig. 6. Outage probability vs episode at the learning stage

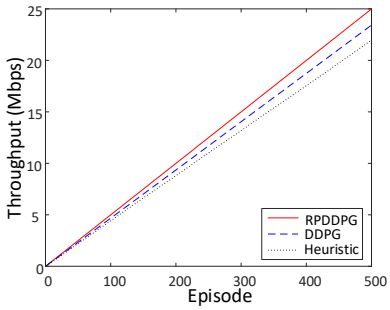

Fig. 7. Throughput vs episode at the operation stage

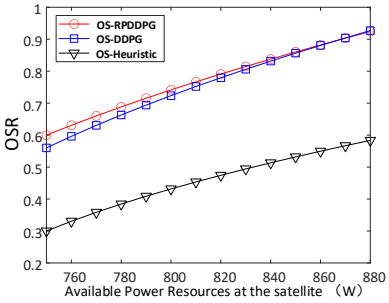

Fig. 8. OSR vs available power resources at the satellite

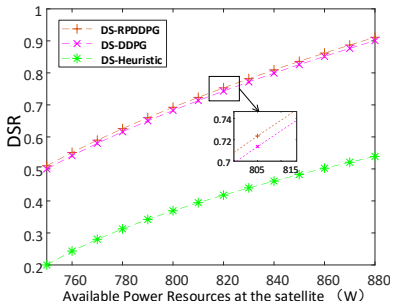

Fig. 9. DSR vs available power resources at the satellite

and random algorithms, and the RPDDPG algorithm always achieves higher rewards (system throughput) compared with the DDPG algorithm. This is because that the RPDDPG reconstructs the action space for the mixed-integer problem by turning a discrete variable to a continuous one with a probabilistic meaning. Meanwhile, the final output integer variables are generated according to the learned probability distribution rather than the simple discretization of continuous variables. In other words, the RPDDPG algorithm can efficiently explore more possibilities in the learning process than the DDPG and better adaptively manage the transmission mode and resource allocation for traffic flows, which indicates that the proposed RPDDPG-based adaptive transmission strategy is effective and efficient in the SAGIN system.

V. Conclusions

In this paper, we have developed the RPDDPG algorithm for the ATSP, which achieves a desired trade-off between the system capacity and the service quality for services. The proposed RPDDPG algorithm can learn a better policy to make joint transmission mode and resource allocation decisions, compared with the DDPG algorithm at the cost of a small amount of storage space. Numerical results demonstrate the effectiveness of the RPDDPG compared with the conventional relaxation-based DDPG algorithm.