Adaptive and Cooperative Resource Scheduling for Satellite-Terrestrial Networks

Yixin Wang†, Di Zhou*†‡, *Member, IEEE,* Min Sheng†, *Senior Member, IEEE,* and Jiandong Li†, *Fellow, IEEE*
†State Key Lab of ISN, Information Science Institute, Xidian University, Xi'an, Shaanxi, 710071, China
‡Science and Technology on Communication Networks Laboratory, Shijiazhuang, China
Email: *zhoudi@xidian.edu.cn

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10000848

***Abstract*—Satellite-terrestrial networks (STNs) consisting of satellite segment and ground segment have been regarded as a desirable solution for 6G. Efficient cooperative resource scheduling strategies, which cover the cooperation in satellite segment for data relay and the cooperation between satellite segment and ground segment for data downloading, play a pivotal role in enhancing the system performance in STNs. Since the dynamic channel condition and energy feeding greatly influence the network status, cooperative resource scheduling should be adaptive to the future environmental fluctuation. In this paper, we model the cooperative resource scheduling problem in STNs as a resource limited Markov Decision Process (MDP). Considering the fact that satellites are unaware of future environmental status, the traditional static optimization solution is infeasible. Therefore, we propose a Deep Reinforcement Learning (DRL) based Cooperative Store-and-Relay Resource Scheduling Algorithm (CSR-RSA), where inter-satellite links are utilized to coordinate with intermittent satellite-ground links for improving the transmission performance of the network. By exploiting the proposed CSR-RSA, the well-trained neural networks can be obtained to generate the adaptive and cooperative resource scheduling strategy without the knowledge of future environmental status. Simulation results verify the effectiveness of the proposed algorithm compared with traditional algorithms.**

***Index Terms*—Satellite-terrestrial networks supported 6G, resource scheduling, environmental adaptability, store and relay mechanism, deep reinforcement learning.**

I. Introduction

Nowadays, satellite-terrestrial networks (STNs) have attracted a great deal of attention in the sixth generation (6G) wireless communication networks. In 6G and future networks, massive Machine-Type Communications (mMTCs), with the goal of supporting the connection for machine-type communication devices (MTCDs) with little human intervention, is assumed to be a typical scenario for the fields of environment monitoring, smart city, etc. [1], [2]. On the one hand, MTCDs may be deployed anywhere. On the other hand, mMTCs should be strong in damage resistance. Therefore, satellites assisted mMTCs is regarded as a suitable construction to provide global coverage for densely deployed MTCDs [3].

Resource scheduling, as an important technology in the field of STNs, will greatly influence the service quality for satellites assisted mMTCs. However, efficient resource scheduling problem still remains to be solved because the practical system is causal, which indicates that the future environmental status with highly dynamic nature is unforeseeable for networks. Therefore, static optimization approaches are inapplicable for the resource scheduling in STNs.

Recent works on resource scheduling of STNs can mainly be divided into two parts, i.e., independent resource scheduling and cooperative resource scheduling. In [4], Zhou *et al.* designed a battery management strategy based on dynamic programming to efficiently balance the mission Quality of Service (QoS) and satellite service lifetime. For scenarios where state transition probabilities are absent, Zhou *et al.* further defined several feature functions and proposed a resource allocation strategy for STNs based on single-agent reinforcement learning to maximize the amount of data transmitted to ground devices [5]. Contrarily, Zhao *et al.* regarded terrestrial devices as agents and proposed an access control and relay selection approach based on distributed Q learning for satellite terrestrial relay networks to maximize the sum rate of the system [6].

For improving the network performance, cooperative resource scheduling is considered. Wu *et al.* focused on the optimization of both power allocation and altitude adjustment in integrated terrestrial-satellite relay networks for maximizing the network throughput, where an aerial relay was introduced to collaboratively provide service for ground devices [7]. Taking concurrent hybrid requests into consideration, Li *et al.* presented a dispersion degree based elastic resource allocation algorithm to alleviate the influence of resource fragmentation on network performance in the software defined satellite optical network [8]. Authors in [9] discussed the data scheduling optimization problem in STNs and presented an improved dynamic programming framework to increase the amount of data offloaded to ground devices.

Aforementioned researches either neglect the cooperation among non-terrestrial segments, or simplify the dynamic channel and onboard resource model in STNs, where statistics knowledge is usually unavailable. In this paper, we focus on the Low Earth Orbit (LEO) satellites assisted mMTCs and

This work was supported in part by the Natural Science Foundation of China under Grant 62121001, Grant U19B2025, and Grant 62001347, in part by Key Research and Development Program of Shaanxi (ProgramNo. 2022ZDLGY05-02), and in part by Young Talent Support Program of Xi'an Association for Science and Technology (No. 095920221337).

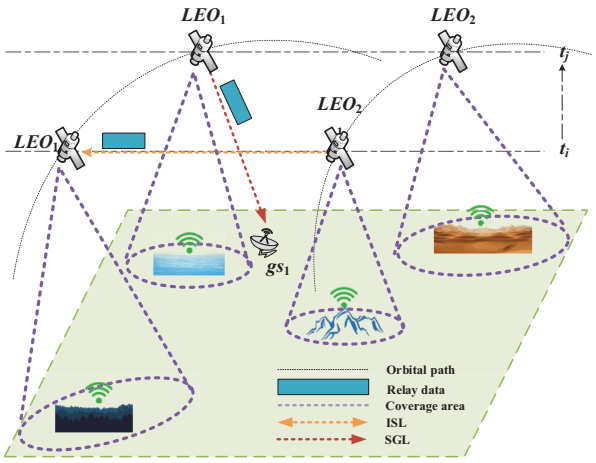


Fig. 1. LEO satellite-terrestrial network.

propose a learning based adaptive and cooperative resource scheduling approach which is free from requiring statistics knowledge. First, we analyze the network peculiarities and model the dynamic channel and onboard resources to formulate the resource evolution. Then, the resource scheduling problem is formulated as an optimization problem for maximizing the amount of data transmitted to ground devices. Solving the problem directly is impracticable because the future dynamic environmental status (i.e., channel condition and energy feeding) cannot be acknowledged in advance. We reformulate the problem as a Markov Decision Process (MDP) and propose a Deep Reinforcement Learning (DRL) based Cooperative Store-and-Relay Resource Scheduling Algorithm (CSR-RSA). Particularly, CSR-RSA, combined with Kuhn–Munkres algorithm, performs resource scheduling according to weighted communication link and power allocation obtained by neural networks. Finally, simulation results show the convergence and verify the effectiveness of the proposed algorithm for time tolerant STNs.

The remainder of this paper is organized as follows. Section II presents the system model and formulates the resource scheduling problem. The proposed resource scheduling algorithm is discussed in Section III. Simulation results are given in Section IV. Finally, Section V concludes the paper.

II. System Model and Problem Formulation

In this section, we model the LEO satellites assisted mMTCs and introduce the channel model and the onboard resource model, which are of great significance in the resource scheduling. Then, we formulate the resource scheduling problem as an optimization problem.

Fig. 1 shows the model of the considered STN. The network consists of several LEO satellites and ground stations (GSs), which are denoted as $LS = \{ls_1, ls_2, \ldots, ls_k, \ldots, ls_N\}$ and $GS = \{gs_1, gs_2, \ldots, gs_l, \ldots, gs_M\}$, respectively. N and M refer to the number of satellites and GSs deployed in the network respectively. Each satellite is equipped with one transmitting antenna and one receiving antenna, which indicates that the satellite offloads its data by either a satellite-to-satellite link (SSL) or a satellite-to-ground link (SGL) and accommodates the relay data for one satellite once.

We divide the continuous network running time into numerous time slots with equal length τ for analytical convenience. We have $\tau = t_{i+1} - t_i$, with the index of time slots $i = 0, 1, \ldots, I-1$. I denotes the number of time slots. Satellites are assumed to take action for cooperative resource scheduling at the beginning of each time slot.

A. Channel model

In this subsection, the dynamic channel model is introduced. As mentioned before, there exist two types of communication links, i.e., SGLs and SSLs. We assume that the channel condition is constant during a time slot. The detailed dynamic channel model can be found in [5].

The achievable data rate (in bps) of an SGL at t_i from ls_k to gs_l is concluded as follows

$$C_i^{(k,l)} = B_c \cdot \log_2\left(1 + SNR_i^{(k,l)}\left(H_i^{(k,l)}, P_i^{(k,l)}\right)\right). \quad (1)$$

Herein, B_c represents the available bandwidth. Besides, $SNR_i^{(k,l)}$ refers to signal-noise ratio of the SGL from ls_k to gs_l at t_i, which relies on the transmission power (in W) $P_i^{(k,l)}$ of ls_k and the channel coefficient $H_i^{(k,l)}$ at t_i.

The satellite collects remote sensing data by turning on onboard sensors, and then the collected data will be offloaded to GSs, where data is processed and extracted for further analysis. However, SGLs do not always exist due to the invisibility resulting from the rotation of both satellites and GSs. Therefore, relaying the data through SSLs before SGLs is an indirect way for helping transmit data to GSs, which also facilitates releasing the cache of satellites to collect more data. In other words, there is no need for satellites to keep the data till SGLs appear.

The achievable data rate (in bps) of an SSL in time slot i from ls_k to ls_j is denoted as

$$C_i^{(k,j)} = H_i^{(k,j)} \cdot P_i^{(k,j)}, \quad (2)$$

where $P_i^{(k,j)}$ and $H_i^{(k,j)}$ are the transmission power (in W) of ls_k and the channel coefficient from ls_k to ls_j at t_i, respectively.

B. Onboard resource model

In this subsection, we propose the dynamic onboard resource model, where multi-dimensional onboard resources are introduced. Since resource scheduling affects the resource evolution, we also formulate the transition of resources while satisfying resource constraints.

1) Battery: Denote the limited battery level of ls_k at t_i as B_i^k. Let $B_{\max}$ be the storage capacity of battery. Since all the behaviors of satellites consume energy, battery energy level is critical during resource scheduling. The battery level of ls_k at t_{i+1} can be calculated as follows

$$B_{i+1}^k = \min\left(B_i^k + S_i^k - E_i^k, B_{\max}\right), \tag{3}$$

where S_i^k and E_i^k is the harvested energy and consumed energy of ls_k in time slot i, respectively.

Generally speaking, energy arrival occurs only when satellites are in the sun phase. In addition, considering that solar rays and the loss of solar panels are random, we denote the battery recharge rate at t_i as SP_i^k (in W) to represent the dynamic and random energy arrival of ls_k, the maximum value of which is denoted as $SP^{\max}$. The energy harvested in time slot i can be calculated as follows

$$S_i^k = SP_i^k \cdot \tau_i^k, \tag{4}$$

where τ_i^k is the duration (in s) when ls_k is exposed to the sun during the time slot i. Note that the energy harvested during the time slot i cannot be utilized until the next time slot $i+1$ reaches.

E_i^k consists of the energy used for mission performing and system maintenance, which is calculated as follows

$$\begin{aligned} E_i^k &= \left(P_r \cdot \tau_r \cdot \sum_{x=1}^{N} \mathbf{M}_{i-1}^{(x,k)}\right) \\ &+ \left(P_c \cdot \tau \cdot \psi_i^k + P_i^{(k,j/l)} \cdot \tau_i^{(k,j/l)}\right) + (P_s \cdot \tau) \\ &= RE_i^k + CE_i^k + E_s \end{aligned}. \tag{5}$$

Herein, P_r is the constant power (in W) for receiving data transmitted from a certain satellite and τ_r is the corresponding receiving time. $\mathbf{M}_{i-1}$ is the match matrix at t_{i-1}, the size of which is $N \times N$ and x is the index of satellite. If ls_x transmits data to ls_k at t_{i-1}, we have $\mathbf{M}_{i-1}^{(x,k)} = 1$. Assume that $\sum_{x=1}^{N} \mathbf{M}_{i-1}^{(x,k)} \leq 1$. P_c is the constant power (in W) for collecting data. ψ_i^k is a binary variable representing whether ls_k turns on its sensor to collect data or not. $P_i^{(k,j/l)}$ represents the transmitting power (in W) at t_i from ls_k to either ls_j or gs_l. $\tau_i^{(k,j/l)}$ refers to the duration time for data transmitting. P_s is the static operation power (in W) for system maintenance. For the sake of brevity, we denote the first term, the second term and the third term as RE_i^k, CE_i^k and E_s, respectively, as is shown in Eq. (5).

Obviously, ψ_i^k and $P_i^{(k,j/l)}$ have a significant effect on future on-board resources and the following constraints in energy should be satisfied when resource scheduling

$$\begin{aligned} RE_i^k &\leq \max\left(0, B_i^k - B_{\min} - E_s\right) \\ CE_i^k &\leq \max\left(0, B_i^k - B_{\min} - E_s - RE_i^k\right) \end{aligned}. \tag{6}$$

Herein, $B_{\min}$ is the safety threshold. Data receiving, collecting and transmitting processes are considered only when the battery level can guarantee the static operation during current slot, among which the data receiving process is considered first.

2) Data buffer: Denote the limited data buffer level of ls_k at t_i as D_i^k. Let $D_{\max}$ be the storage capacity of data buffer. The data buffer level of ls_k at t_{i+1} can be calculated as follows

$$D_{i+1}^k = \min\left(D_i^k + SD_i^k - TD_i^{(k,j/l)}, D_{\max}\right), \tag{7}$$

where SD_i^k represents the amount of data (in Bit) stored in the data buffer during time slot i and $TD_i^{(k,j/l)}$ is the amount of data (in Bit) transmitted in the time slot i. $TD_i^{(k,j/l)}$ can be expressed as

$$TD_i^{(k,j/l)} = C_i^{(k,j/l)} \cdot \tau_i^{(k,j/l)}. \tag{8}$$

In addition, SD_i^k can be written as

$$SD_i^k = \chi_r \cdot \tau_r \cdot \sum_{x=1}^{N} \mathbf{M}_{i-1}^{(x,k)} + \chi_c \cdot \tau \cdot \psi_i^k, \tag{9}$$

in which χ_r and χ_c denote the receiving rate (in bps) and the collecting rate (in bps), respectively. Similarly, the following constraint in data is required to hold when resource scheduling

$$TD_i^{(k,j/l)} \leq D_i^k. \tag{10}$$

C. Problem formulation

In the proposed resource scheduling problem, our goal is to maximize the amount of data transmitted to GSs while satisfying resources constraints. We first formulate the resource scheduling problem as an optimization problem, which is expressed as follows

$$\begin{aligned} \max\ & \sum_{i=0}^{I-1} \sum_{k=1}^{N} TD_i^{(k,l)}, \forall gs_l \in GS \\ \text{s.t.}\ & (6), (10) \\ & B_i^k \leq B_{\max} \\ & D_i^k \leq D_{\max} \\ & \forall i, k \end{aligned}. \tag{11}$$

However, the proposed optimization problem cannot be solved directly since satellites can only restore the past and present information, where environment information (i.e., channel condition and energy arrival) of global time sequence is absent.

III. Proposed Resource Scheduling Algorithm

In this section, we reformulate the resource scheduling problem as a resource limited Markov Decision Process (MDP). Further, since the system is causal, DRL structure is adopted for the adaptation of resource scheduling and we present a DRL based resource scheduling algorithm without the requirement of global statistics knowledge.

A. Basics of DRL derived resource scheduling

We adopt the MDP model to formulate the resource scheduling process, where state space, action space, and reward are introduced. At the beginning of each time slot t_i, each satellite observes the state and selects the action. Then, the state evolves until the beginning of next time slot t_{i+1} comes. Here are some key elements of MDP.

1) State Space S: State at t_i is denoted as S_i, which can be expressed as $S_i = \left\{S_i^1, S_i^2, \cdots, S_i^k, \cdots, S_i^N\right\}$. S_i^k can be

Algorithm 1 CSR-RSA

Input:
Time variant network topology, channel condition and energy arrival;
Output:
The online Q network;

Initialization: exploration rate ε, learning rate α, the online Q network parameters ϑ, the target Q network parameters ϑ' , batch size, replay memory, empty interval I_e, copy interval I_c, the initial state S_{init}.
for training episode $e = 0, 1, \cdots, e^{\mathrm{MAX}} - 1$ **do**
  Start a new episode;
  if $e\%I_e = 0$ **then**
    Empty the replay memory and initialize the state $S_i \Leftarrow S_{init}$;
  end if
  for time slot $i = 0, 1, \cdots, i^{\mathrm{MAX}} - 1$ **do**
    Observe the state S_i and update the exploration rate ε according to Eq. (15);
    Choose an action A_i according to Algorithm 2;
    Perform the state evolution and obtain the next state S_{i+1} according to Eqs. (3), (7);
    for satellites that transmit data to GSs **do**
      Distribute the reward;
    end for
    Store the sample $\left(S_i^k, a_i^k, R_i^k, S_{i+1}^k\right), \forall ls_k \in LS$ into replay memory;
    if the number of samples is not less than the batch size **then**
      Select a batch of samples and feed them into the network for training ϑ;
    end if
  end for
  if $e\%I_c = 0$ **then**
    $\vartheta' = \vartheta$;
  end if
end for

further denoted as $S_i^k = \left\{B_i^k, D_i^k, SP_i^k, H_i^{(k,j/l)}\right\}$. B_i^k and D_i^k represent the battery level and data buffer level of ls_k at t_i, respectively. SP_i^k is the battery recharge rate and $H_i^{(k,j/l)}$ is the channel coefficient at t_i.

2) Action Space A: The network collaborative action at t_i is denoted as $A_i = \left\{a_i^1, a_i^2, \cdots, a_i^k, \cdots, a_i^N, \mathbf{M}_i\right\}$. For each a_i^k, we have $a_i^k = \left\{\psi_i^k, P_i^{(k,j/l)}\right\}$. Note that $\mathbf{M}_i$ determines the topology match of SSLs at t_i.

3) Reward R: The amount of data that ls_k transmits to the gs_l at t_i is regarded as the total reward (in Bit) R_i^k, which can be expressed as follows

$$R_i^k = TD_i^{(k,l)}. \tag{12}$$

Actions that help to relay the data should be rewarded because they contribute to the reward R_i^k in an indirect way.

Algorithm 2 Act Selection Policy

Input:
The state S_i, the exploration rate ε;
Output:
the action A_i;

Generate a random number n belonging to $[0, 1]$;
if $n < \varepsilon$ **then**
  for satellites that are accessible to GSs **do**
    Calculate the feasible actions and randomly select an action;
  end for
  for satellites that are inaccessible to GSs **do**
    Calculate the feasible actions and randomly select an action, where the connected satellite will not be attachable to other satellites;
  end for
else
  for satellites that are accessible to GSs **do**
    Calculate the feasible actions and select the action that maximizes action value;
  end for
  for satellites that are inaccessible to GSs **do**
    For each possible connection, calculate the feasible actions and select the action that maximizes action value;
  end for
  Utilize the Kuhn–Munkres Algorithm to determine the optimal match;
  Select the actions that correspond to the optimal match;
end if

Considering that the state space is continuous and infinite and its elements are causal, we utilize DRL to solve the resource scheduling problem, where Deep Q-Network (DQN) is adopted. By continuous iteration, DQN maximizes the expected accumulative reward, which can also be represented by the action value $Q_\pi(S, A)$ with the policy π [10]. The loss function is defined as

$$\begin{aligned} L_i(\vartheta) = \mathrm{E}\Big[\Big(R_i + \gamma \max_A Q(S_{i+1}, A; \vartheta') \\ - Q(S_i, A_i; \vartheta)\Big)^2\Big], \end{aligned} \tag{13}$$

where γ is the discount rate. ϑ and ϑ' are the online network parameters and target network parameters, respectively. For updating online network parameters, the following gradient calculation equation is introduced

$$\begin{aligned} \nabla_\vartheta L_i(\vartheta) = \mathrm{E}\big[\nabla_\vartheta Q(S_i, A_i; \vartheta) \cdot \\ \Big(R_i + \gamma \max_A Q(S_{i+1}, A; \vartheta') - Q(S_i, A_i; \vartheta)\Big)\Big]. \end{aligned} \tag{14}$$

The target network copies the parameters of the online network to achieve its parameter updating.

B. Cooperative store-and-relay resource scheduling algorithm

We propose a Cooperative Store-and-Relay Resource Scheduling Algorithm (CSR-RSA) based on DRL to achieve adaptive and cooperative resource scheduling, which is shown in Algorithm 1. At the beginning of each slot, network observes the state and updates the exploration rate ε. Let ε be updated by the following rule

$$\varepsilon_i = 0.9999^{e \cdot i^{\mathrm{MAX}} + i}. \tag{15}$$

Then, as is shown in Algorithm 2, each satellite calculates the feasible actions according to the resource constraints shown in Eqs. (6), (10). In the case of $n < \varepsilon$, satellites randomly select the action among their feasible actions. When $n \geq \varepsilon$, satellites exploit the experience and choose the action that maximizes the action value. Note that the action value can be obtained by feeding the state into the online Q network. Guided by the ε, the agent has the probability to randomly choose an action, which helps to avoid local optimization.

We assume that GSs can receive data transmitted from several satellites simultaneously. Therefore, satellites that are accessible to GSs determine the power allocation scheme in satellite-to-ground process without link conflict. However, since satellites are equipped with single input antenna, link conflict may occur in inter-satellite relay process. Hence, satellites that are inaccessible to GSs should collaboratively determine the power allocation and inter-satellite topology match scheme to avoid link conflict. We adopt the Kuhn–Munkres algorithm to obtain the optimal network matching [11]. Specifically, under each available ISL, satellite obtains the optimal action that maximizes action value, which is regarded as the current link weight. Then, we can obtain a weighted match table with the size of $N \times N$, based on which the optimal topology match can be found by utilizing the Kuhn–Munkres algorithm. Finally, we can get the optimal action A_i that maximizes the accumulative action value.

Taking the channel diversity into consideration, both satellite-to-satellite scenario and satellite-to-ground scenario have their respective online Q network, target Q network, and replay memory. During the process of training, the network parameters are iteratively updated according to gradient calculation equation. When the algorithm ends, we can obtain the final online network parameters for future resource scheduling.

IV. Simulation Results

In this section, we conduct the simulation results to verify the effectiveness of the proposed resource scheduling algorithm. Notably, topology and energy arrival are obtained by STK and Matlab. Our simulation is achieved by Python Simulators.

The considered STN consists of 6 GSs and 6 LEO satellites. In this paper, we evaluate the algorithm performance from the perspective of the amount of episode downlink data (AEDD), which is defined as the amount of data transmitted from satellites to GSs during an episode. To demonstrate the effectiveness of the proposed algorithm, we design three traditional algorithms, i.e., Q learning based conservative resource scheduling algorithm (CRSA), selfish and greedy resource scheduling algorithm (SGRSA), and aimless resource scheduling algorithm (ARSA).

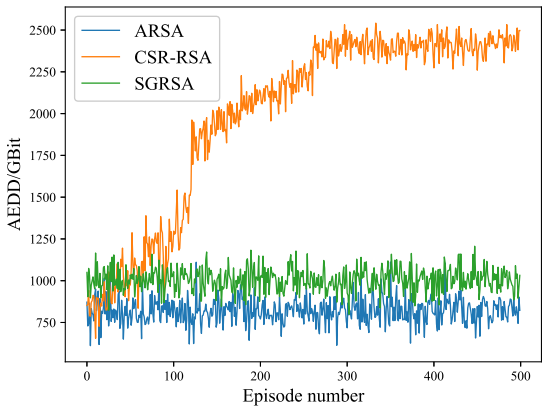
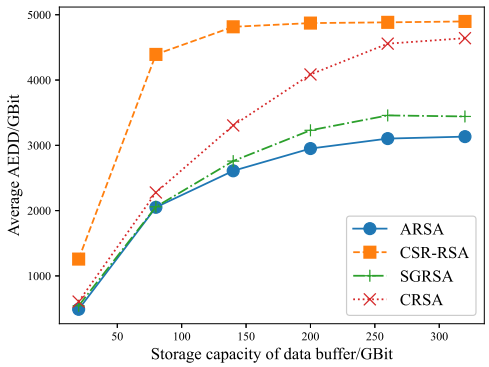


Fig. 2. AEDD *v.s.* Episode number.

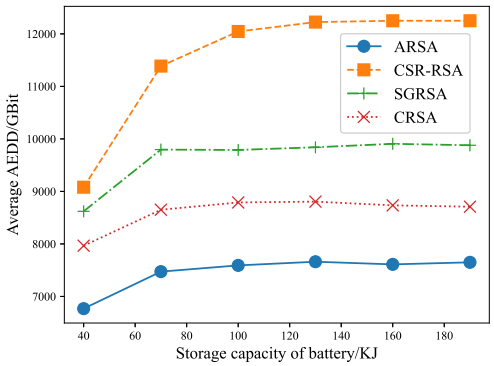
- CRSA: In the CRSA, the satellite keeps the collected data until the SGL appears. The scheme has no concept of collaboration and the data downloading process solely relies on SGLs.
- SGRSA: In the SGRSA, each satellite makes full use of the energy to collect the data as much as possible. Therefore, the SGRSA is deficient in collaboration since the satellite always thinks highly of its own profit.
- ARSA: The ARSA takes the inter satellite collaboration into consideration and schedules resource randomly without preference. In other words, the ARSA ignores the resource condition and schedules the resource with no objective.

Fig. 2 shows the convergence of the CSR-RSA from the perspective of the amount of data transmitted to GSs. We can observe that the AEDD increases first and stabilizes finally. This can be explained by the tradeoff between exploring and exploiting. At the early stage of training, the network prefers scheduling resources in a random way to collect abundant and diverse samples. With the evolution of training, the network tends to learn from the experience for achieving better resource scheduling.

Further, we study the impact of resource settings on network performance as shown in Fig. 3. It can be seen that with the increasing storage capacity, the average AEDD shows a non-declining trend. Specifically, Fig. 3(a) is obtained with the resource limitation of $B_{\max} = 70\,(\mathrm{KJ}), SP^{\max} = 30\,(\mathrm{W})$. When storage capacity of data buffer is relatively small, the network performance is limited by the short buffer resource. However, when storage capacity of data buffer reaches to a high level, the network performance is constrained by the energy resource. Therefore, the increasing storage



(a) Average AEDD *v.s.* Storage capacity of data buffer.



(b) Average AEDD *v.s.* Storage capacity of battery.

Fig. 3. Average AEDD with different storage capacity.

capacity of data buffer improves the network performance first, and then the network performance hits a bottleneck and becomes stable. Herein, the proposed CSR-RSA makes the best contribution to the average AEDD due to its store-and-relay mechanism and intelligent cooperation. ARSA performs cooperative resource scheduling in a random way, which leads to the worst performance. With the resource limitation of $D_{\max} = 120\,(\mathrm{GBit})\,, SP^{\max} = 70\,(\mathrm{W})$, Fig. 3(b) shows the trend of average AEDD under various storage capacity of battery. It can be seen that increasing storage capacity of battery is efficient in improving the network performance only when the storage capacity is in certain value range. When the storage capacity is relatively large, the network performance is limited by other dimensional resources. Therefore, it is useless to steadily improve the network performance by greedily increasing the storage capacity. Similarly, the proposed CSR-RSA outperforms the other schemes and ARSA is the worst scheme. Based on Q learning, CRSA discretizes the network state and ignores the satellite cooperation and thus the performance is erratic. SGRSA puts self-interest first and ignores the accumulative network interest, which deteriorates the network performance.

V. Conclusion

This paper focused on satellites assisted mMTCs and designed an intelligent resource scheduling strategy which is adaptive and cooperative for time tolerant STNs. Specifically, we adopted the dynamic channel model and elaborated on the dynamic onboard resource model to describe the resource evolution process while satisfying resource constraints. Considering that the future statistics knowledge is unknown for the system, we formulated the adaptive and cooperative resource scheduling as a resource limited MDP for the feasibility. Then we introduced the DRL framework and proposed a Kuhn–Munkres algorithm integrated CSR-RSA to guide the power allocation and topology match for maximizing the accumulative amount of data offloaded to GSs. Finally, simulation results showed that CSR-RSA can efficiently improve the network performance compared with traditional resource scheduling algorithms under dynamic resource settings.

References

[1] W. Kim, Y. Ahn, and B. Shim, "Deep neural network-based active user detection for grant-free NOMA systems," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2143–2155, Apr. 2020.

[2] Y. Wang, T. Wang, Z. Yang, D. Wang, and J. Cheng, "Throughput-oriented non-orthogonal random access scheme for massive MTC networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1777–1793, Mar. 2020.

[3] S. Kota and G. Giambene, "6G integrated non-terrestrial networks: Emerging technologies and challenges," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, Montreal, QC, Canada, Jun. 2021, pp. 1–6.

[4] D. Zhou, M. Sheng, Y. Zhu, J. Li, and Z. Han, "Mission QoS and satellite service lifetime tradeoff in remote sensing satellite networks," *IEEE Wireless Communications Letters*, vol. 9, no. 7, pp. 990–994, Mar. 2020.

[5] D. Zhou, M. Sheng, Y. Wang, J. Li, and Z. Han, "Machine learning-based resource allocation in satellite networks supporting internet of remote things," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6606–6621, Oct. 2021.

[6] B. Zhao, G. Ren, X. Dong, and H. Zhang, "Distributed Q-learning based joint relay selection and access control scheme for IoT-oriented satellite terrestrial relay networks," *IEEE Communications Letters*, vol. 25, no. 6, pp. 1901–1905, Jun. 2021.

[7] B. Wu, F. Fang, and S. Fu, "Improving the performance of terrestrial-satellite relay networks by configuring aerial relay," *IEEE Transactions on Vehicular Technology*, early access 2021.

[8] Y. Li, Q. Zhang, R. Gao, X. Xin, H. Yao, F. Tian, and M. Guizani, "An elastic resource allocation algorithm based on dispersion degree for hybrid requests in satellite optical networks," *IEEE Internet of Things Journal*, early access 2021.

[9] D. Zhou, M. Sheng, J. Luo, R. Liu, J. Li, and Z. Han, "Collaborative data scheduling with joint forward and backward induction in small satellite networks," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3443–3456, May 2019.

[10] X. Tang, J. Chen, T. Liu, Y. Qin, and D. Cao, "Distributed deep reinforcement learning-based energy and emission management strategy for hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, Oct. 2021.

[11] Y. Liu, X. Fang, and M. Xiao, "Joint transmission reception point selection and resource allocation for energy-efficient millimeter-wave communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 412–428, Jan. 2021.