Energy-Constrained Online Scheduling for Satellite-Terrestrial Integrated Networks

Xin Gao, *Graduate Student Member, IEEE,* Jingye Wang, Xi Huang, *Member, IEEE,* Qiuyu Leng, Ziyu Shao*, *Senior Member, IEEE,* Yang Yang, *Fellow, IEEE*

Abstract—In satellite-terrestrial integrated networks, it is a common practice to schedule real-time tasks from low Earth orbit (LEO) satellites to ground stations (GSs) for data processing. However, the joint task scheduling and resource allocation under unknown environment dynamics (*e.g.*, transmission latency) remains to be a challenging problem. First, the tradeoff between task latencies and energy consumption should be carefully considered when making decisions to minimize task latencies under time-averaged energy consumption constraints. Second, to learn the environment uncertainties and minimize the system performance loss (*i.e.*, regret) in terms of task latencies, both online feedback and offline history should be leveraged efficiently, and the accompanying exploration-exploitation tradeoff should be dealt with in a proper way. In this article, we formulate the joint task scheduling and resource allocation problem as a constrained combinatorial multi-armed bandit (CMAB) problem. To solve the problem, by integrating online learning, online control, and offline historical information, we propose a *Task scheduling and Resource allocation scheme with Data-driven Bandit Learning* called *TRDBL*. Our theoretical and numerical results show that TRDBL achieves a sublinear time-averaged regret while satisfying the time-averaged energy consumption constraints.

Index Terms—Satellite-terrestrial integrated networks, task scheduling, resource allocation, data-driven bandit learning, learning-aided online control.

1 Introduction

In recent years, satellite-terrestrial integrated networks have attracted great attention due to the rapid development of low Earth orbit (LEO) satellite systems [2]. By far, more and more LEO satellites have been launched for real-time Earth observation tasks such as intelligence reconnaissance, natural disaster surveillance, and environment monitoring [3] [4]. To complete such tasks, LEO satellites collect Earth observation data with onboard sensors and transmit the collected data to ground stations (GSs) for further data processing [5].

Considering that each LEO satellite may have access to multiple GSs simultaneously [2], a key issue for each LEO satellite is how to schedule observation tasks to proper GSs in an online fashion to achieve minimum task latencies. However, due to the rapid satellite movement and the changeable weather condition, the qualities of wireless channels between LEO satellites and GSs change frequently over time [6] [7]. As a result, the instantaneous transmission latencies from LEO satellites to GSs are hard to be measured in real time. To this end, online learning should be integrated into the task scheduling process to infer the statistics of transmission latencies implicitly based on system feedback. Meanwhile, to improve the learning accuracies, the pre-maintained offline historical information about transmission latencies can be exploited to aid the online learning procedure [8]. Besides, the decision making for task scheduling and resource allocation often involves the concern of energy efficiency [9] [10], which requires online control on the energy consumption to aid the decision making process.

Towards such an integrated design which combines online control, online learning, and offline historical information, there are three challenges to be addressed. The *first* is concerning the non-trivial tradeoff between task latencies and energy consumptions in the online control procedure. Generally, to reduce the task processing latencies on GSs, more computing resources are required, resulting in higher energy consumptions. Nonetheless, due to the scarcity of energy, the time-averaged budget constraints on energy consumptions should be satisfied along with the objective of minimizing task latencies. To satisfy the requirements towards both metrics, the tradeoff between task latencies and energy consumptions should be carefully tackled in the online control procedure. The *second* challenge lies in the online learning procedure. When an LEO satellite schedules tasks to GSs, it can either *exploit* current knowledge by

- *Xin Gao is with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China, also with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China, and with the University of Chinese Academy of Sciences, Beijing 100049, China. (E-mail: gaoxin@shanghaitech.edu.cn)*
- *Jingye Wang, Qiuyu Leng and Ziyu Shao are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China. (E-mail: {wangjy5, lengqy, shaozy}@shanghaitech.edu.cn)*
- *Xi Huang is with Shenzhen Institute of Artificial Intelligence and Robotics for society (AIRS), Shenzhen 518100, China. Prior to that, he was with School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China. (E-mail: huangxi@shanghaitech.edu.cn)*
- *Yang Yang is with Shanghai Institute of Fog Computing Technology (SHIFT), ShanghaiTech University, Shanghai 201210, China, also with the Research Center for Network Communication, Peng Cheng Laboratory, Shenzhen 518000, China, and also with Shenzhen SmartCity Technology Development Group Company Ltd., Shenzhen 518046, China (E-mail: yangyang@shanghaitech.edu.cn).*

*This work was partially supported by Nature Science Foundation of Shanghai under Grant 19ZR1433900. Partial results in this work have been published in IEEE International Conference on Communications (ICC), 2021 [1]. (*Corresponding author: Ziyu Shao)*

selecting GSs with empirically low estimated transmission latencies to achieve a better short-term performance; or it can *explore* new knowledge by selecting GSs with less feedbacks to improve the long-term performance. Such an exploration-exploitation dilemma should be carefully addressed to achieve a high learning efficiency. Moreover, the offline history information should be properly incorporated to improve the learning accuracies. The *third* challenge is towards the interplay between online learning and online control procedures. Specifically, the online learning procedure, if conducted ineffectively, may vitiate the online control procedure and incur performance loss; meanwhile, the online control procedure, if carried out improperly, would lead to a low learning efficiency and excessive energy consumptions.

In this article, we address all the aforementioned challenges. In particular, we studied the problem of task scheduling and resource allocation in satellite-terrestrial networks with unknown environment and offline historical information under time-averaged energy consumption constraints. By investigating such a problem from the perspective of Combinatorial Multi-Armed Bandit (CMAB) [11], we propose an online scheme that integrates online control, online learning, and offline historical information with a performance guarantee. The contributions and key results of our work are summarized as follows.

- **Problem Formulation:** We formulate the task scheduling and resource allocation problem as a stochastic optimization problem under uncertainties. The goal of the problem is to minimize the total task latencies while satisfying the time-averaged energy consumption constraints. By exploiting the structure of the formulated problem, we reformulate it as a constrained combinatorial multi-armed bandit (CMAB) problem.
- **Algorithm Design:** We propose a scheme called *TRDBL* (Task scheduling and Resource allocation with Data-driven Bandit Learning) to solve the formulated problem. TRDBL is composed of an online learning procedure and an online control procedure. In the online learning procedure, the data-driven bandit learning technique [17] is adopted to estimate the transmission latencies from LEO satellites to GSs by leveraging both the offline historical information and online feedback information. In the online control procedure, the Lyapunov optimization technique [18] is applied to determine the task scheduling and resource allocation decisions based on the estimations provided by the online learning procedure.
- **Theoretical Analysis:** Our theoretical analysis shows that TRDBL satisfies all the time-averaged energy consumption constraints and achieves a time-averaged regret of $O(1/V + 1/T + \sqrt{(\log T)/(T + H_{\min})})$ over the finite horizon T. Here V is a positive parameter that can be tuned, and $H_{\min}$ is the minimal number of offline historical observations among GSs.
- **Numerical Evaluation:** We conduct extensive simulations to evaluate the performance of TRDBL and its variants. The simulation result shows that our proposed scheme can reduce task latency effectively under time-averaged energy consumption constraints.
- **New Degree of Freedom for Network Design:** We investigate the benefits of offline historical information in the design of satellite-terrestrial integrated networks via both theoretical analysis and numerical evaluations. Our results provide novel insights to network designers to improve the performance of satellite-terrestrial integrated networks.

The rest of this article is organized as follows. Section 2 discusses the related works. Section 3 introduces our system model and problem formulation. Section 4 elaborates our algorithm and Section 5 provides theoretical performance analysis for the algorithm. Section 6 evaluates our algorithm through simulations. Finally, Section 7 concludes this work.

2 Related Work

Satellite-terrestrial integrated network has recently attracted much attention as it enables the provision of different technologies such as traffic offloading [10], global connectivity [19], data caching [20], and Earth observation [21]. In the scenario of Earth observation, efforts with different goals have been made to improve the performance of satellite-terrestrial networks. For example, Wang *et al.* [4] studied the multi-resource coordinate scheduling problem in a satellite-terrestrial integrated network to maximize the sum priorities of successfully scheduled tasks. In [12], by utilizing the offloading between satellites, Zhang *et al.* minimized the energy consumption of data transmission from satellites to ground and maximized the network throughput while satisfying the delay requirement of tasks. By assuming static task arrivals, these works apply offline control techniques such as graph theory and iterative optimization to solve their problems. However, in practice, tasks generally arrive in a dynamic way and the future task arrivals are hard to be accurately predicted in satellite-terrestrial integrated networks. As for works that consider dynamic task arrivals, those who are most related to our work are generally conducted from two different perspectives: the online control perspective and online learning perspective.

Works Based on Online Control: Works carried out from the online control perspective adopt techniques such as the Lyapunov optimization method to deal with the dynamic environment settings and conduct online scheduling. For example, faced with dynamic task arrivals to satellites, Wang *et al.* [13] maximized the network throughput approximately by scheduling the inter-satellite and satellite-to-ground contacts in an online manner. Huang *et al.* [14] proposed a dynamic power allocation approach to maximize the network throughput and minimize the energy consumption. In [15], they further took the admission control and task dispatch into consideration when making decisions. He *et al.* [16] designed a task scheduling scheme which jointly determines the scheduling period and allocates antenna time block to task arrivals with the goal of maximizing the number of completed tasks under unpredicted task arrivals. Although the effectiveness of these solutions, they assumed that the instantaneous network environment dynamics such as wireless transmission conditions are observable when

TABLE 1. Comparison between our work and related works

	Optimization Metrics	Dynamic Arrivals	Offline Control	Online Control	Online Learning	Offline Historical Information
[4]	Sum priorities of successfully scheduled tasks		•			
[12]	Throughput & energy consumption & transmission latency		•			
[13]	Throughput	•		•		
[14]	Throughput & energy consumption	•		•		
[15]	Throughput & energy consumption	•		•		
[16]	Number of completed tasks	•		•		
[9]	Task delay & energy consumption & server usage cost				•	
Our Work	Task latency & energy consumption	•		•	•	•

making decisions. However, in practice, such information is usually unknown a priori.

Works Based on Online Learning: Works conducted from the online learning perspective consider the case when the instantaneous network environment dynamics are unknown or their distributions are hard to be modeled accurately. These works adopt online learning techniques to deal with the uncertainties. For example, Cheng *et al.* [9] employed deep reinforcement learning techniques to propose a task scheduling and resource allocation scheme. However, their solution cannot deal with time-averaged constraints and does not exploit the benefits of offline historical information.

Novelty of Our Work: In our work, we deal with both unknown network environment dynamics and time-averaged constraints. Moreover, we exploit the effectiveness of offline historical information to aid our online learning procedure. By integrating Lyapunov optimization based online control, CMAB based online learning, and offline historical information, we jointly optimize the task scheduling and resource allocation in satellite-terrestrial integrated networks with rigorous theoretical performance guarantee. The comparison between our work and existing works is shown in Table 1.

3 System Model

In this Section, we first introduce our model in detail, and then formulate a stochastic task scheduling and resource allocation problem under time-averaged energy consumption constraints. Our key notations are listed in Table 2.

TABLE 2. Key notations

Notation	Description
$\mathcal{M}$	Set of all GSs with $\lvert\mathcal{M}\rvert \triangleq M$
$\mathcal{M}(t)$	Set of accessible GSs in time slot t, $\mathcal{M}(t) \subset \mathcal{M}$
$\mathcal{A}(t)$	Set of task arrivals in time slot t with $\lvert\mathcal{A}(t)\rvert \triangleq N(t)$ and $\mathcal{A}(t) = \{A_1(t), \ldots, A_{N(t)}(t)\}$
$S_n(t)$	Observation data size of task $A_n(t)$
$I_{n,m}(t)$	Indicator of whether task $A_n(t)$ is scheduled to GS m
$F_{n,m}(t)$	CPU cycle frequency allocated to task $A_n(t)$ on GS m
$W_m(t)$	Unit transmission latency from the LEO satellite to GS m in time slot t with mean $w_m \triangleq \mathbb{E}[W_m(t)]$
$W_m^h(k)$	Instantaneous unit transmission latency recorded by the k-th offline historical observation
H_m	Number of offline historical observations available for unit transmission latency from LEO satellite to GS m
$\tilde{w}_m(t)$	Estimated mean unit transmission latency from LEO satellite to GS m in time slot t
$L_{n,m}(t)$	Number of CPU cycles needed to process per bit of task $A_n(t)$ on GS m
$D_n^{tr}(t)$	Transmission latency of task $A_n(t)$
$D_n^{pr}(t)$	Processing latency of task $A_n(t)$
$D_n(t)$	Total latency of task $A_n(t)$
$C_m(t)$	Unit transmission energy consumption on LEO satellite when transmitting data to GS m in time slot t
$E^{tr}(t)$	Transmission energy consumption on LEO satellite in time slot t
$E_m^{pr}(t)$	Processing energy consumption on GS m in time slot t
κ_m	Effective switched capacitance related to the chip architecture on GS m
γ_0	Energy budget on LEO satellite for data transmission
γ_m	Energy budget on GS m for data proccessing

3.1 Basic Model

We consider a satellite-terrestrial integrated network which operates on a slotted time horizon and the time slots are indexed by $\{0, 1, 2, \ldots\}$. Particularly, we consider the interplay between one low Earth orbit (LEO) satellite and M ground stations (GSs) which are denoted by set $\mathcal{M} \triangleq \{1, 2, \ldots, M\}$. We show our system model in Figure 1. In each time slot t, the LEO satellite can only access to a subset of GSs which are in the line-of-sight of the LEO satellite. Due to the movement of the LEO satellite, the connections between the LEO satellite and GSs change over time, resulting in a time-varying accessible GS subset [4] [6]. We denote such an accessible GS subset as $\mathcal{M}(t)$. We consider the case where the widely applied Orthogonal Frequency Division Multiple Access (OFDMA) technology [22] [23] is employed to support simultaneous access from the LEO satellite to multiple GSs. Besides, we assume that the co-channel interference is eliminated through interference management techniques such as zero-forcing beamforming [24] [25].

3.2 System Workflow

The workflow of the system during each time slot t is shown as follows. At the beginning of the time slot, a set of tasks $\mathcal{A}(t) \triangleq \{A_1(t), A_2(t), \ldots, A_{N(t)}(t)\}$ arrive, where $N(t)$ ($0 \leq N(t) \leq n_{\max}$) is the number of tasks and it varies over time slots. For each task $A_n(t)$, an observation data of size $S_n(t)$ ($0 < S_n(t) \leq s_{\max}$) is collected by the onboard sensors of the LEO satellite. To complete the task, the LEO satellite needs to *schedule* the task to one of the GSs in set $\mathcal{M}(t)$ by transmitting the corresponding observation data

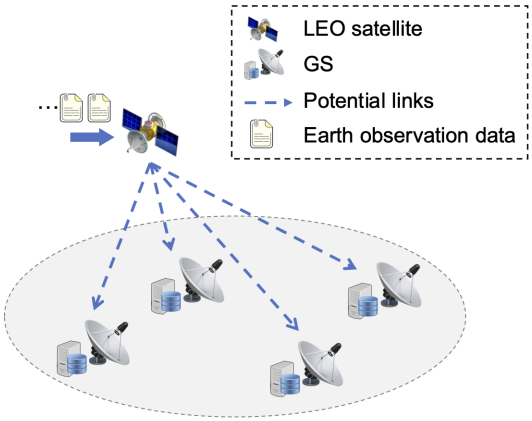


Fig. 1. An illustration of satellite-terrestrial integrated networks.

to the selected GS. Once receiving the data, the selected GS sends the information about task transmission latency back to the LEO satellite, and then *allocates* its computing resources to processing the received data. We assume that the whole procedure is completed within the time slot.

3.3 Decision Variables

3.3.1 Task Scheduling Decision

We denote the task scheduling decision in time slot t as $\boldsymbol{I}(t) \triangleq (I_{n,m}(t))_{n\in\mathcal{N}(t),m\in\mathcal{M}(t)}$, in which $\mathcal{N}(t) \triangleq \{1,2,\ldots,N(t)\}$. Each entry $I_{n,m}(t) = 1$ if task $A_n(t)$ is scheduled to GS m and $I_{n,m}(t) = 0$ otherwise. Since each task should be scheduled to one and only one GS in the accessible GS subset $\mathcal{M}(t)$, the following constraints must be satisfied:

$$\sum_{m\in\mathcal{M}(t)} I_{n,m}(t) = 1,\ \forall n \in \mathcal{N}(t), t, \tag{1}$$

$$I_{n,m}(t) \in \{0,1\},\ \forall n \in \mathcal{N}(t), m \in \mathcal{M}(t), t, \tag{2}$$

$$I_{n,m}(t) = 0,\ \forall n \in \mathcal{N}(t), m \notin \mathcal{M}(t), t. \tag{3}$$

3.3.2 Resource Allocation Decision

In our model, we adopt the CPU cycle frequency as the dominant computing resource[1]. We assume that each GS has the DVFS (Dynamic Voltage and Frequency Scaling) [26] capability and can adjust its CPU cycle frequency. We denote the CPU cycle frequency allocated to task $A_n(t)$ on GS m as $F_{n,m}(t)$, then it should satisfy

$$f_{\min} \le F_{n,m}(t) \le f_{\max},\quad \forall n \in \mathcal{N}(t), m \in \{m' | I_{n,m'}(t) = 1\}, t, \tag{4}$$

$$F_{n,m}(t) = 0,\ \forall n \in \mathcal{N}(t), m \in \{m' | I_{n,m'}(t) = 0\}, t, \tag{5}$$

where $f_{\min}$ and $f_{\max}$ are pre-given constants that satisfy $0 < f_{\min} \le f_{\max}$. Constraints in (4) ensure that the computing resource allocated to each task is bounded. Constraints in (5) state that a GS will not allocate any computing resource to tasks that are not scheduled to it.

1. Our scheme can be also extended to take other kinds of resources into consideration.

3.4 Performance Metrics

3.4.1 Task Latency

Latency is one of the key QoS measurements in satellite-terrestrial integrated networks [27]. For each task $A_n(t) \in \mathcal{A}(t)$, its latency is composed of the transmission latency from the LEO satellite to its allocated GS and the processing latency on the GS.

Transmission Latency: If task $A_n(t)$ is scheduled to GS m, it will experience a transmission latency of $S_n(t)W_m(t)$, where $W_m(t)$ ($w_{\min} \le W_m(t) \le w_{\max}$) is the unit transmission latency from the LEO satellite to GS m during time slot t. Therefore, under task allocation decision $\boldsymbol{I}(t)$, the transmission latency of task $A_n(t)$ is given by

$$D_n^{tr}(t) \triangleq \hat{D}_{n,t}^{tr}(\boldsymbol{I}(t)) = \sum_{m\in\mathcal{M}(t)} S_n(t) W_m(t) I_{n,m}(t). \tag{6}$$

The unit transmission latency $W_m(t)$ is a random variable with an unknown mean w_m and it is assumed to be *i.i.d.* across time slots. We further assume that the data transmissions from the LEO satellite to GSs can be finished within the current time slot. Therefore, the feedback information about transmission latencies can be sent back to the LEO satellite within current time slot.

The LEO satellite retains a number of offline historical observations about the transmission latencies from it to GSs [8]. Specifically, the offline historical observations about the unit transmission latency to GS $m \in \mathcal{M}$ are denoted by a sequence $\{W_m^h(0), W_m^h(1), \ldots, W_m^h(H_m - 1)\}$, where $H_m \ge 0$ is the length of the sequence, *i.e.*, the number of offline historical observations. When $H_m = 0$, there is no offline historical observation. In particular, each observation $W_m^h(k)$ is assumed to have the same distribution as the unit transmission latency $W_m(t)$.

Processing Latency: To characterize the processing latency of task $A_n(t)$ on GS m, we define $L_{n,m}(t)$ ($0 \le L_{n,m}(t) \le l_{\max}$) as the number of CPU cycles needed to process per *bit* of task $A_n(t)$ on GS m. Accordingly, the processing latency is $L_{n,m}(t)S_n(t)/F_{n,m}(t)$. Then given task scheduling decision $\boldsymbol{I}(t)$ and CPU cycle frequency allocation $\boldsymbol{F}(t)$ ($\boldsymbol{F}(t) \triangleq (F_{n,m}(t))_{n\in\mathcal{N}(t),m\in\mathcal{M}}$), the processing latency of task $A_n(t)$ is

$$D_n^{pr}(t) \triangleq \hat{D}_{n,t}^{pr}(\boldsymbol{I}(t), \boldsymbol{F}(t)) = \sum_{m\in\mathcal{M}(t)} L_{n,m}(t) S_n(t) I_{n,m}(t) / F_{n,m}(t). \tag{7}$$

Therefore, the total latency of task $A_n(t)$ is

$$D_n(t) \triangleq \hat{D}_{n,t}(\boldsymbol{I}(t), \boldsymbol{F}(t)) = \hat{D}_{n,t}^{tr}(\boldsymbol{I}(t)) + \hat{D}_{n,t}^{pr}(\boldsymbol{I}(t), \boldsymbol{F}(t)), \tag{8}$$

i.e., the sum of transmission latency and processing latency.

3.4.2 Energy Consumption

The energy consumption for each task is composed of the transmission energy consumption on the LEO satellite and the processing energy consumption on the GS that the task is scheduled to.

Transmission Energy: The LEO satellite consumes energy when it transmits data to GSs. We denote the energy

consumption of transmitting per bit of data to GS m in time slot t as $C_m(t)$ ($0 < C_m(t) \leq c_{\max}$). Then given task scheduling decision $\boldsymbol{I}(t)$, the total transmission energy consumption on LEO satellite in time slot t is

$$E^{tr}(t) \triangleq \hat{E}_t^{tr}(\boldsymbol{I}(t)) = \sum_{n\in\mathcal{N}(t)} \sum_{m\in\mathcal{M}(t)} C_m(t) S_n(t) I_{n,m}(t). \quad (9)$$

As the LEO satellite can only harvest energy from the solar and store the energy in the rechargeable battery [28], the energy consumption on the LEO satellite should be carefully controlled. To this end, we consider the following long-term time-averaged energy constraint on the LEO satellite:

$$\limsup_{T'\to\infty} \frac{1}{T'} \sum_{t=0}^{T'-1} \mathbb{E}\left[E^{tr}(t)\right] \leq \gamma_0, \quad (10)$$

where $\gamma_0 > 0$ is a predefined energy budget on the LEO satellite for data transmission. Note that $\frac{1}{T'}\sum_{t=0}^{T'-1}\mathbb{E}[E^{tr}(t)]$ is the average of expected energy consumption on the LEO satellite over time slots. It is defined as the time-averaged energy consumption on the LEO satellite.

Processing Energy: Each GS $m \in \mathcal{M}(t)$ consumes energy when processing tasks that are scheduled to it. Given the task scheduling decision $\boldsymbol{I}(t)$ and CPU cycle frequency allocation $\boldsymbol{F}(t)$, according to [29], the total energy consumption of GS m for data processing in time slot t is

$$E_m^{pr}(t) \triangleq \hat{E}_{m,t}^{pr}(\boldsymbol{I}(t), \boldsymbol{F}(t)) = \sum_{n\in\mathcal{N}(t)} \kappa_m L_{n,m}(t) S_n(t) I_{n,m}(t) (F_{n,m}(t))^2. \quad (11)$$

Here κ_m is the effective switched capacitance which is related to the chip architecture and it is measurable in practice [30]. Regarding the scarcity of energy and the avenue for service providers, we consider the following long-term time-averaged constraint on the processing energy consumption for each GS:

$$\limsup_{T'\to\infty} \frac{1}{T'} \sum_{t=0}^{T'-1} \mathbb{E}\left[E_m^{pr}(t)\right] \leq \gamma_m, \ \forall m \in \mathcal{M}. \quad (12)$$

Here $\gamma_m > 0$ is the pre-given energy budget on GS m for data processing.

3.5 Problem Formulation

Our goal is to minimize the total task latency over T time slots under long-term time-averaged energy consumption constraints on the LEO satellite and GSs. Specifically, our problem formulation is shown as follows:

$$\underset{\{\boldsymbol{I}(t),\boldsymbol{F}(t)\}_t}{\text{minimize}} \sum_{t=0}^{T-1} \sum_{n\in\mathcal{N}(t)} \mathbb{E}\left[D_n(t)\right] \quad (13a)$$

$$\text{subject to } (1)(2)(3)(4)(5)(10)(12). \quad (13b)$$

4 Algorithm Design

For problem (13), if the instantaneous unit transmission latency $W_m(t)$ from the LEO satellite to each GS $m \in \mathcal{M}(t)$ is observable when making decisions in each time slot t, it can be solved asymptotically optimally by adopting online control techniques such as Lyapunov optimization [18]. However, such information is usually not available in practice. To solve for problem (13) under uncertainties, we employ online learning techniques to estimate the mean unit transmission latencies $\{w_m\}_{m\in\mathcal{M}}$ based on both offline historical and online feedback information. In the procedure of online learning, a key issue to be tackled is to deal with the dilemma between exploration (gain new knowledge) and exploitation (leverage current knowledge). More specifically, in each time slot, when scheduling tasks, the LEO satellite can either select GSs who are rarely selected to gain new knowledge about their latency performance, or select GSs who have the empirically lowest unit transmission latencies to achieve a better short-term performance.

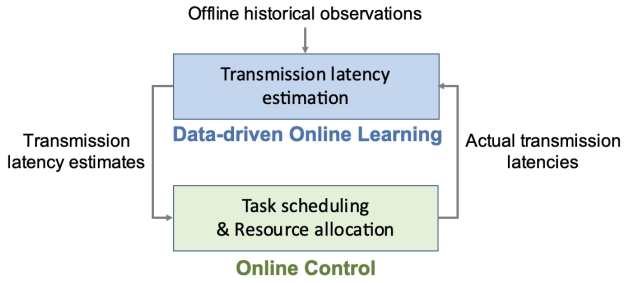


Fig. 2. An illustration of our algorithm design.

In our work, we reformulate problem (13) as a constrained combinatorial multi-armed bandit (CMAB) problem, and adopt data-driven bandit learning to deal with the exploration-exploitation dilemma and estimate uncertainties in the online learning procedure. Based on the estimates, we employ Lyapunov optimization in the online control procedure to ensure the energy consumption constraints while minimizing the total task latency. With an effective integration of online learning, online control, and offline historical information, we propose a joint task scheduling and resource allocation scheme called TRDBL (Task scheduling and Resource allocation with Data-driven Bandit Learning) to solve for the reformulated CMAB problem. Figure 2 shows the flow of our algorithm. In each time slot, under TRDBL, the LEO satellite first estimates the transmission latency to each GS, and then schedules tasks to available GSs based on the estimates. After receiving data from the LEO satellite, each GS sends feedback information about the true transmission latencies to the LEO satellite, and allocates computing resources to the tasks scheduled to it.

In the following Subsections, we first reformulate the original problem (13) as a constrained CMAB problem, and then introduce the details of our algorithm design with respect to online learning and online control procedures, respectively. Finally, we discuss the computational complexity of our algorithm.

4.1 Problem Reformulation

The classical CMAB [11] considers a sequential game between an agent and a bandit with multiple arms (*a.k.a.* actions). The game is played over a finite number of rounds. In each round, the agent selects a subset of arms to play,

then the environment reveals a reward for each played arm. The objective of the agent is to gain the largest expected cumulative reward. However, there is a fundamental challenge that the expected reward brought by each arm is unknown to the agent. Li *et al.* [31] extended the model of classical CMAB problem by assuming that each arm may be unavailable in some rounds and considering arm fairness constraints. Inspired by their work, we reformulated our latency minimization problem (13) as a constrained CMAB problem with extended settings.

To reformulate problem (13), we view the LEO satellite as the agent and GSs as arms. In each time slot t, the LEO satellite (agent) schedules tasks to GSs (arms) in the available set $\mathcal{M}(t)$. Note that more than one task may be scheduled to the same GS during the time slot. If task $A_n(t)$ is scheduled to GS m (arm m is played for task $A_n(t)$), a reward of $R_n(t) \triangleq -D_n(t)$ would be returned to the LEO satellite. Note that the task latency $D_n(t)$ defined in (8) is determined by not only the task scheduling decision $I_{n,m}(t)$ but also the resource allocation decision $F_{n,m}(t)$. With such a reformulation, problem (13) can be rewritten as the following equivalent problem:

$$\underset{\{\boldsymbol{I}(t),\boldsymbol{F}(t)\}_t}{\text{maximize}} \quad \sum_{t=0}^{T-1} \sum_{n\in\mathcal{N}(t)} \mathbb{E}\left[R_n\left(t\right)\right] \tag{14a}$$

$$\text{subject to} \quad (1)(2)(3)(4)(5)(10)(12). \tag{14b}$$

By the definition of reward $R_n(t) \triangleq -D_n(t)$, to maximize the total reward is equivalent to minimize the total task latency, *i.e.*, problem (14) is equivalent to problem (13) under the formulated CMAB model.

Remark: Though our constrained CMAB model is motivated by the proposed model in [31], the settings of our model have some major differences from the settings in [31]. First, since multiple tasks can be transmitted to the same GS in each time slot, each arm may be played for more than one time during one time slot. Second, the energy consumption constraints (10) and (12) in our problem are more complex than the arm fairness constraints in [31]. The fairness of each arm is only determined by the selection frequency of this arm. However, the energy consumptions on LEO and GSs are determined by selections of different arms. As a result, the selections of all arms are coupled together under our energy consumption constraints. The final extension is that we consider a more general reward model which is a function of both the arm selection on LEO satellite and the computing resource allocations on GSs.

To characterize the performance loss due to the decision making under uncertainties, we define the *regret* over T time slots as

$$Reg\left(T\right) \triangleq \frac{1}{T}\left(R^*(T) - \sum_{t=0}^{T-1} \sum_{n\in\mathcal{N}(t)} \mathbb{E}\left[R_n\left(t\right)\right]\right), \tag{15}$$

where $R^*(T)$ is the optimal cumulative expected reward that can be achieved. The reward maximization problem (14) is then equivalent to the following regret minimization problem:

$$\underset{\{\boldsymbol{I}(t),\boldsymbol{F}(t)\}_t}{\text{maximize}} \quad Reg\left(T\right) \tag{16a}$$

$$\text{subject to} \quad (1)(2)(3)(4)(5)(10)(12). \tag{16b}$$

In the following subsections, we integrate online control, online learning, and offline historical information to solve problem (16).

4.2 Data-Driven Online Learning Procedure

In the online learning procedure, we employ data-driven bandit learning to deal with the uncertainties. In each time slot t, after data transmissions from the LEO satellite to GS $m \in \mathcal{M}$, the feedback information about instantaneous unit transmission latency $W_m(t)$ from the LEO satellite to GS m will be transmitted back to the LEO satellite. Based on the collected online feedback information, and in the meanwhile leveraging the retained offline historical information, the mean unit transmission latency w_m to GS m is estimated as follows when $t \geq 1$:

$$\tilde{w}_m\left(t\right) = \max\left\{\bar{w}_m\left(t\right) - \omega_0\sqrt{\frac{3\log t}{2\left(H_m + h_m\left(t\right)\right)}}, w_{\min}\right\}, \tag{17}$$

where $\omega_0 \triangleq w_{\max} - w_{\min}$. When $t = 0$, we estimate w_m as $\tilde{w}_m(t) = w_{\min}$. Here $\bar{w}_m(t)$ is the empirical mean of the unit transmission latency to GS m based on both offline historical information and collected online feedback information. The counter $h_m(t)$ counts the number of previous time slots (before time slot t) during which GS m is selected. When $t = 0$, we initialize $\bar{w}_m(t) = \sum_{k=0}^{H_m-1} W_m^h(k)$ and $h_m(t) = 0$. When $t \geq 1$, they are defined respectively as follows:

$$\bar{w}_m(t) \triangleq \left(\sum_{k=0}^{H_m-1} W_m^h(k) + \sum_{\tau=0}^{t-1} W_m(\tau) \prod_{n\in\mathcal{N}(\tau)} I_{n,m}(\tau)\right) / (H_m + h_m(t)), \tag{18}$$

$$h_m\left(t\right) \triangleq \sum_{\tau=0}^{t-1} \prod_{n\in\mathcal{N}(\tau)} I_{n,m}\left(\tau\right). \tag{19}$$

Remark 1: In estimation (17), the term $\bar{w}_m(t)$ reflects the acquired knowledge about the mean transmission latency and thus it is known as the *exploitation* term. The term $\omega_0\sqrt{\frac{3\log t}{2(H_m+h_m(t))}}$ is called the *confidence radius* [32]. It represents our measure of how the empirical mean $\bar{w}_m(t)$ is close to the true mean w_m, and a larger confidence radius indicates more uncertainty in $\bar{w}_m(t)$. The confidence radius provides an adaptive exploration for the online learning procedure. Specifically, in the case when the true value of mean unit transmission latency w_m from the LEO satellite to GS m is under-explored (*i.e.*, $H_m + h_m(t) \ll t$), the confidence radius for $\bar{w}_m(t)$ becomes large, resulting in a small latency estimate $\tilde{w}_m(t)$ by (17). As a result, there is a high probability that GS m will be selected by the LEO satellite in time slot t since the goal of task scheduling is to minimize task latencies. On the contrary, when w_m is sufficiently explored, the confidence radius is small and GS m will only be selected if it has a sufficiently good empirical performance (*i.e.*, a sufficiently small $\bar{w}_m(t)$). Therefore, the confidence radius is also called the *exploration* term. Other learning methods such as ε-greedy [33] can also be applied in our algorithm design, which is discussed in Section 6.

Remark 2: The confidence radius in (17) also characterizes the interaction of offline historical information and online

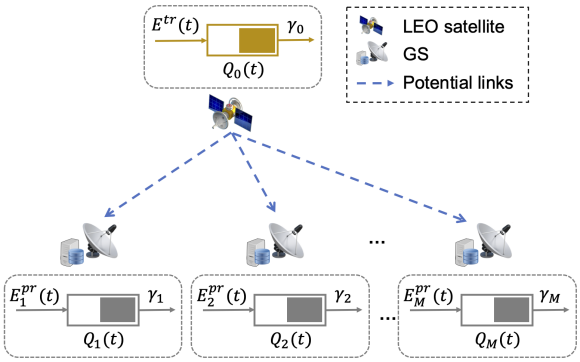


Fig. 3. An illustration of virtual queues. The LEO satellite maintains a virtual queue $Q_0(t)$ with an input of $E^{tr}(t)$ and an output of γ_0, and each GS $m \in \mathcal{M}$ maintains a virtual queue $Q_m(t)$ with an input of $E_m^{pr}(t)$ and an output of γ_m. If the queueing process $\{Q_m(t)\}_t$ is strongly stable, then the corresponding time-averaged energy consumption cost constraint can be satisfied.

feedback information. In the early stage when t is much smaller than H_m, we have $H_m \gg h_m(t)$ and the estimate is mainly determined by the offline historical observations. However, as time goes by, more online feedbacks are collected and the impact of online feedback information on the estimate grows gradually.

4.3 Online Control Procedure

In the online control procedure, we adopt Lyapunov optimization [18] to solve the regret minimization problem under time-averaged constraints. To handle the energy consumption constraints in (10) and (12), we equivalently transform them into the queue stability constraints by introducing the virtual queueing processes $\{Q_0(t)\}_t$ for the LEO satellite and $\{Q_m(t)\}_t$ for each GS $m \in \mathcal{M}$. The backlogs of these virtual queues are initialized to be zero and updated in each time slot $t \geq 0$ as follows:

$$Q_0(t+1) = [Q_0(t) - \gamma_0]^+ + E^{tr}(t), \tag{20}$$

$$Q_m(t+1) = [Q_m(t) - \gamma_m]^+ + E_m^{pr}(t), \ \forall m \in \mathcal{M}, \tag{21}$$

in which $[\cdot]^+ \triangleq \max\{\cdot, 0\}$. We illustrate these virtual queues in Figure 3. For each constraint in (10) and (12), it will be satisfied when the corresponding queueing process is strongly stable [18]. An intuitive understanding for the transformation is that to guarantee the stability of the virtual queue, we should ensure that the mean queue input (*i.e.*, energy consumption) is no larger than the mean queue output (*i.e.*, energy budget). Based on the above analysis, problem (16) can be equivalently transformed to the following problem:

$$\underset{\{\boldsymbol{I}(t),\boldsymbol{F}(t)\}_t}{\text{maximize}} \ \mathbb{E}[Reg(T)] \tag{22a}$$

$$\text{subject to} \ \ (1)(2)(3)(4)(5), \tag{22b}$$

$$\limsup_{T'\to\infty} \frac{1}{T'} \sum_{t=0}^{T'-1} \mathbb{E}[Q_m(t)] < \infty, \quad \forall m \in \{0\} \cup \mathcal{M}. \tag{22c}$$

By adopting the Lyapunov optimization technique [18], we solve for problem (22) approximately by solving for the following sub-problem in each time slot t:

$$\underset{\boldsymbol{I}(t),\boldsymbol{F}(t)}{\text{minimize}} \ \sum_{n\in\mathcal{N}(t)} \sum_{m\in\mathcal{M}} \tilde{p}_{n,m}^t(F_{n,m}(t)) I_{n,m}(t) \tag{23a}$$

$$\text{subject to} \ \ (1)(2)(3)(4)(5). \tag{23b}$$

Here $\tilde{p}_{n,m}^t(F_{n,m}(t))$ represents the price of scheduling task $A_n(t)$ to GS m. It is a function of $F_{n,m}(t)$ and is defined as follows:

$$\tilde{p}_{n,m}^t(F_{n,m}(t)) \triangleq VS_n(t)(\tilde{w}_m(t) + L_{n,m}(t)/F_{n,m}(t)) + S_n(t)(Q_0(t)C_m(t) + Q_m(t)\kappa_m L_{n,m}(t)(F_{n,m}(t))^2), \tag{24}$$

where V is a positive parameter which can be tuned. The detailed derivation is shown in Appendix A.

Note that problem (23) is a mixed integer programming problem. Its optimal solution is given as follows[2]. For each $n \in \mathcal{N}(t)$ and $m \in \mathcal{M}(t)$, we have

$$I_{n,m}(t) = \begin{cases} 1, & \text{if } m = \arg\min_{m'\in\mathcal{M}(t)} \tilde{p}_{n,m'}^t(\check{F}_{n,m'}(t)); \\ 0, & \text{otherwise;} \end{cases} \tag{25}$$

$$F_{n,m}(t) = \begin{cases} \check{F}_{n,m}(t), & \text{if } I_{n,m}(t) = 1; \\ 0, & \text{otherwise;} \end{cases} \tag{26}$$

where $\check{F}_{n,m}(t)$ is defined as

$$\check{F}_{n,m}(t) = \max\{\min\{\sqrt[3]{V/(2\kappa_m Q_m(t))}, f_{\max}\}, f_{\min}\}. \tag{27}$$

The pseudocode of our proposed task scheduling and resource allocation scheme TRDBL is shown in Algorithm 1. We provide the following remarks for TRDBL.

Remark 1: As shown in (24) and (27), the value of V determines the relative importance of minimizing task latency compared to satisfying energy consumption constraints. Under a large value of V, the first term $VS_n(t)(\tilde{w}_m(t) + L_{n,m}(t)/F_{n,m}(t))$ (*i.e.*, estimated task latency multiplied by weight V) in (24) becomes the major term. Then under TRDBL, the LEO satellite will schedule task $A_n(t)$ to the GS with the minimal estimated latency $S_n(t)(\tilde{w}_m(t) + L_{n,m}(t)/F_{n,m}(t))$. Moreover, with a larger value of V, according to (27), GSs will allocated higher CPU cycle frequencies to tasks scheduled to them to reduce task processing latencies.

Remark 2: TRDBL ensures the energy consumption constraints (10) and (12) by avoiding the appearance of large virtual queue backlog sizes. Note that the virtual queue backlog size can be viewed as a counter which keeps track of the part of energy consumption in excess of the budget. When the time-averaged energy consumption of the LEO satellite or a GS tends to violate the budget, its corresponding virtual queue backlog increases to be large. Under the circumstances, for the LEO satellite, it will select GSs with small transmission energy consumptions according to (24). For each GS, on one hand, it tends to be not selected by the LEO satellite because of its large virtual queue backlog size according to (24); on the other hand, it will allocate low CPU cycle frequencies to tasks scheduled to it according to (27).

2. If more than one GS achieves $\min_{m'\in\mathcal{M}(t)} \tilde{p}_{n,m'}(\check{F}_{n,m'}(t))$, the LEO satellite would select one of such GSs uniformly at random.

Algorithm 1 Task scheduling and Resource allocation with Data-driven Bandit Learning (TRDBL)

1: Initialize $h_m(0) = 0$, $\bar{w}_m(0) = \frac{1}{H_m}\sum_{k=0}^{H_m-1} W_m^h(k)$ and $\tilde{w}_m(0) = w_{\min}$ for each GS $m \in \mathcal{M}$. In each time slot $t \in \{0, 1, \cdots, T-1\}$:
%Latency Estimation
2: **for** each GS $m \in \mathcal{M}$ **do**
3: **if** $h_m(t) > 0$ **then**
4: $\tilde{w}_m(t) \leftarrow \max\left\{\bar{w}_m(t) - \omega_0\sqrt{\frac{3\log t}{2(h_m(t)+H_m)}}, w_{\min}\right\}$.
5: **end if**
6: **end for**
%Task Scheduling
7: Initialize task scheduling decision $I_{n,m}(t) = 0$ for each $n \in \mathcal{N}(t)$ and $m \in \mathcal{M}$.
8: **for** each task $A_n(t) \in \mathcal{A}(t)$ **do**
9: $\mathcal{M}_{\min}(t) \leftarrow \{m | m \in \arg\min_{m' \in \mathcal{M}(t)} \tilde{p}_{n,m'}^t(\check{F}_{n,m'}(t))\}$.
10: Uniformly randomly selects GS m from set $\mathcal{M}_{\min}(t)$ and set $I_{n,m}(t) \leftarrow 1$.
11: **end for**
12: The LEO satellite schedules tasks in set $\mathcal{A}(t)$ to GSs according to decision $\boldsymbol{I}(t)$.
%Computing Resource Allocation
13: Initialize computing resource allocation $F_{n,m}(t) = 0$ for each $n \in \mathcal{N}(t)$ and $m \in \mathcal{M}$.
14: **for** each GS $m \in \mathcal{M}(t)$ **do**
15: Set $F_{n,m}(t) \leftarrow \check{F}_{n,m}(t)$ for each task $A_n(t) \in \mathcal{A}(t)$ that is scheduled to GS m, *i.e.*, task $A_n(t)$ with $I_{n,m}(t) = 1$.
16: **end for**
17: Each GS $m \in \mathcal{M}(t)$ processes received tasks according to computing resource allocation $\{F_{n,m}(t)\}_{n\in\mathcal{N}(t)}$ on it.
%Update of Virtual Queues and Statistics
18: Update virtual queues $\boldsymbol{Q}(t)$ according to (20) and (21).
19: **for** each GS $m \in \mathcal{M}$ **do**
20: $h_m(t+1) \leftarrow h_m(t) + \prod_{n\in\mathcal{N}(t)} I_{n,m}(t)$.
21: $\bar{w}_m(t+1) \leftarrow \frac{h_m(t)+H_m}{h_m(t+1)+H_m}\bar{w}_m(t) + \frac{W_n(t)\prod_{n\in\mathcal{N}(t)} I_{n,m}(t)}{h_m(t+1)+H_m}$.
22: **end for**

Remark 3: In TRDBL, the online control and online learning procedures are combined in the following way. On one hand, the online control procedure leverages the estimates for transmission latencies obtained in the online learning procedure to aid its decision makings for task scheduling. On the other hand, the online learning procedure uses the received feedback information under the task scheduling decision determined in the online control procedure, together with the retained offline historical information, to improve its estimation accuracies. In this way, the two procedures are effectively integrated to achieve the goal of minimizing task latencies under time-averaged energy consumption constraints.

4.4 Computational Complexity of TRDBL

There are four processes in TRDBL: latency estimation (lines 2-6), task scheduling (lines 7-11), computing resource allocation (lines 13-16), and update of virtual queues and statistics (lines 18-22). The computational complexity of TRDBL is mainly incurred by the task scheduling process. Specifically, in the worst case, the for-loop (lines 8-11) would involve $n_{\max}$ iterations. Note that the computational complexity of line 9 is $O(M)$. Accordingly, the computational complexity of TRDBL is $O(n_{\max}M)$.

5 Performance Analysis

5.1 Energy Consumption Constraints

An energy budget vector $\boldsymbol{\gamma} \triangleq (\gamma_0, \gamma_1, \ldots, \gamma_M)$ is said to be *feasible* if there exists a feasible solution to problem (13) given the energy budget vector $\boldsymbol{\gamma}$. The set of all feasible energy budget vectors is defined as the *maximal feasibility region* of problem (13). Based on above definitions, we provide the following theorem to show the feasibility of TRDBL in terms of the energy consumption constraints.

Theorem 1. Suppose that the energy budget vector $\boldsymbol{\gamma}$ lies in the interior of the maximal feasibility region of problem (13), then TRDBL satisfies the time-averaged energy constraints in (10) and (12). More specifically, the virtual queueing processes $\{Q_m(t)\}_{m\in\{0\}\cup\mathcal{M},t}$ defined in (20) and (21) are strongly stable and satisfy

$$\limsup_{T'\to\infty}\frac{1}{T'}\sum_{t=0}^{T'-1}\sum_{m\in\{0\}\cup\mathcal{M}}\mathbb{E}[Q_m(t)] \le \Gamma/\epsilon + Vs_{\max}n_{\max}(w_{\max} + l_{\max}/f_{\min})/\epsilon, \quad (28)$$

where ϵ is a positive constant which satisfies that $\boldsymbol{\gamma} - \epsilon\mathbf{1}$ ($\mathbf{1}$ is the $(M+1)$-dimensional all-ones vector) still lies in the interior of the maximal feasibility region. The parameter Γ is defined as $\Gamma \triangleq n_{\max}^2 s_{\max}^2(c_{\max}^2 + l_{\max}^2 f_{\max}^4\sum_{m\in\mathcal{M}}\kappa_m^2)/2 + \sum_{m\in\mathcal{M}\cup\{0\}}\gamma_m^2/2$. The constant $n_{\max}$ is the maximal number of task arrivals in each time slot, $s_{\max}$ is the maximal data size of each task, $w_{\max}$ is the maximal unit transmission latency, $l_{\max}$ is the maximal number of CPU cycles needed to process per bit of a task, κ_m is a positive and measurable constant related to the processing energy consumption on GS m, $f_{\min}$ and $f_{\max}$ are the minimal and maximal CPU cycle frequencies could be allocated to each task.

The proof of Theorem 1 is given in Appendix B.

Remark 1: Theorem 1 shows that TRDBL is feasible to problem (13) when $\boldsymbol{\gamma}$ is an interior point of the maximal feasibility region of the problem. More specifically, by ensuring that all virtual queues are strongly stable, TRDBL satisfies all the energy consumption constraints. Moreover, the time-averaged total virtual queue backlog size is positively proportional to the value of parameter V.

5.2 Regret Bound

The theoretical upper bound for the regret achieved by TRDBL is characterized by the following theorem.

Theorem 2. Under TRDBL, the regret $Reg(T)$ defined in (15) has the following upper bound:

$$Reg(T) \le \frac{\Gamma}{V} + \frac{4\omega_0 n_{\max}s_{\max}}{T} + 2\omega_0 n_{\max}s_{\max}\sqrt{\frac{6n_{\max}M\log T}{T+H_{\min}}}, \quad (29)$$

where $H_{\min} \triangleq \min_{m\in\mathcal{M}} H_m$, $\omega_0 \triangleq w_{\max} - w_{\min}$, $w_{\min}$ and $w_{\max}$ are the minimal and maximal unit transmission latencies. Other parameters are defined exactly the same as in Theorem 1.

The complete proof of Theorem 2 is given in Appendix C.

Remark 2-1: The first term Γ/V in the right-hand side of (29) is induced by the online control procedure of TRDBL: To satisfy the energy consumption constraints, TRDBL sometimes sacrifices the reward (task latency) by making decisions which consume a small amount of energy but may bring high task latencies. We note that such a term is inversely proportional to the value of V. With a larger value of V, TRDBL focuses more on reducing task latencies rather than ensuring energy consumption constraints, and hence achieves a lower regret. Moreover, the term Γ/V approaches zero with a sufficiently large value of V. In real systems, the selection of the value of V depends on the design tradeoff of the systems.

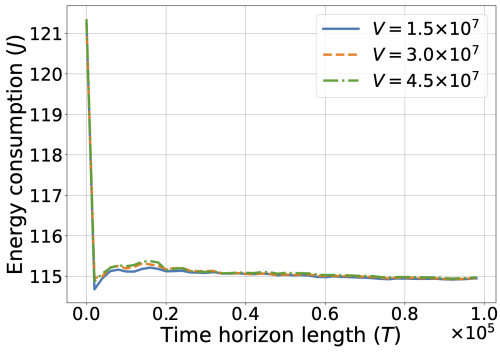
Remark 2-2: The last two terms are $O(1/T + \sqrt{(\log T)/(T + H_{\min})})$. They are incurred by the data-driven online learning procedure. As the value of horizon length T increases, these terms reduce and approach zero, thus the regret bound approaches γ/V. Next, we consider the regret upper bound under different values of $H_{\min}$ when the horizon length T is fixed. Particularly, when $H_{\min} = 0$, our problem degenerates to the case when there is no offline historical information and the regret bound is $O(1/V + \sqrt{(\log T)/T})$. When $H_{\min} > 0$ (*i.e.*, the offline historical information is available), the regret bound reduces with the increase of the value of $H_{\min}$. In the following, we consider the regret bound under four cases:

1) When $H_{\min} = O(1)$, *i.e.*, a constant value unrelated to T. In this case, when compared to the case without offline historical information, the value of the regret upper bound in (29) reduces, but its order remains to be $O(1/V + \sqrt{(\log T)/T})$.
2) When $H_{\min} = \Theta(T)$, *i.e.*, the number of offline historical observations is comparable to T. In this case, the regret bound reduces while its order is still $O(1/V + \sqrt{(\log T)/T})$.
3) When $H_{\min} = \Theta(T \log T)$, the order of the regret bound approaches $O(1/V + \sqrt{1/T})$ under a sufficiently large value of T.
4) When $H_{\min} = \Theta(T^2 \log T)$, there is sufficient offline historical information such that the LEO satellite can estimate transmission latencies accurately when the system begins. In this case, the second term of the regret bound in (29) becomes the dominant one when compared to the last term. As a result, the order of the regret bound reduces to $O(1/V + 1/T)$.

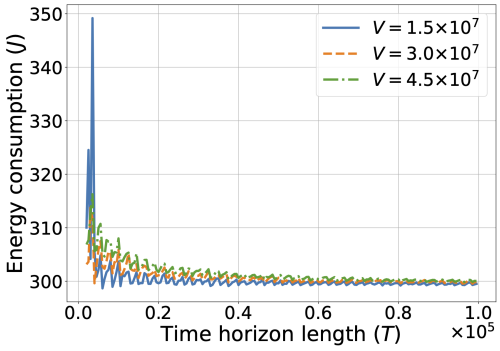
6 Numerical Results

6.1 Simulation Settings

Our parameter settings in simulations are based on the commonly adopted settings in satellite-terrestrial integrated systems [4] [6]. Specifically, we consider the interplay between one LEO satellite and 36 GSs. In each time slot, due to the dynamic characteristic of the network, only 5 GSs are accessible to the LEO satellite. The length of each time slot is set to be 3 minutes. In each time slot t, the number $N(t)$ of task arrivals is sampled from a Poisson distribution with parameter $\lambda = 6$, while the size of each task (in the unit of *bits*) is generated from a uniform distribution over the interval $(10^7, 10^8)$. The transmission latency (in the unit of *s/bit*) for each GS m is generated from a clipped Gaussian distribution over the interval $(R_{\min}(m), R_{\max}(m))$ with a mean of $(R_{\min}(m) + R_{\max}(m))/2$, where $R_{\min}(m)$ and $R_{\max}(m)$ are the minimum and the maximum transmission latencies which are sampled uniformly from the intervals $(2 \times 10^{-9}, 2 \times 10^{-8})$ and $(2 \times 10^{-8}, 2 \times 10^{-7})$, respectively. As for the unit transmission energy consumption $C_m(t)$ (in the unit of *J/bit*), it is sampled from a uniform distribution over the interval $(2 \times 10^{-7}, 5 \times 10^{-7})$. We set the minimum and the maximum CPU cycle frequencies on each GS as $f_{\min} = 5 \times 10^8$ cycles/s and $f_{\max} = 5 \times 10^9$ cycles/s, respectively. As for the parameter setting related to processing energy consumption on each GS m, we set the effective switched capacitance in (11) as $\kappa_m = 10^{-24}$ W·s^3/cycle^3. For energy constraints, by adopting the settings in [34], we set $\gamma_0 = 115$ *J* for the LEO satellite and $\gamma_m = 300$ *J* for each GS $m \in \{1, 2, \ldots, 36\}$. Unless otherwise stated, the time horizon length T is fixed as 10^5 time slots and the number of offline historical observations H_m is set to be 1000 for each GS m (*i.e.*, $H_{\min} = 1000$). All of the experimental results are averaged over 10 different random seeds.



(a) Energy consumption on the LEO satellite.



(b) Energy consumption on GS 10.

Fig. 4. Energy consumption over time.

6.2 Performance Evaluation

6.2.1 Guarantee of Energy Consumption Constraints

We take the LEO satellite and the 10-th GS (GS 10) as examples to illustrate the time-averaged energy consumptions under TRDBL given different values of V ($V \in \{1.5 \times 10^7, 3.0 \times 10^7, 4.5 \times 10^7\}$) in Figure 4. As the time horizon length T increases, the time-averaged energy consumptions on the LEO satellite (see Figures 4(a)) and GS 10 (see

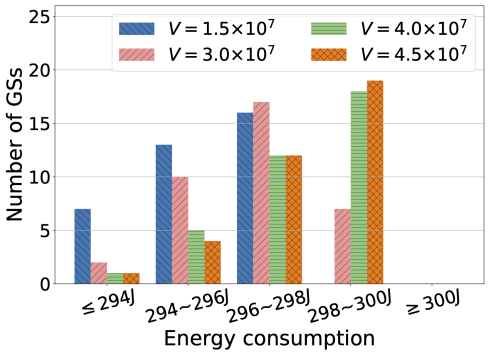


Fig. 5. Energy consumptions on all GSs given $T = 10^5$.

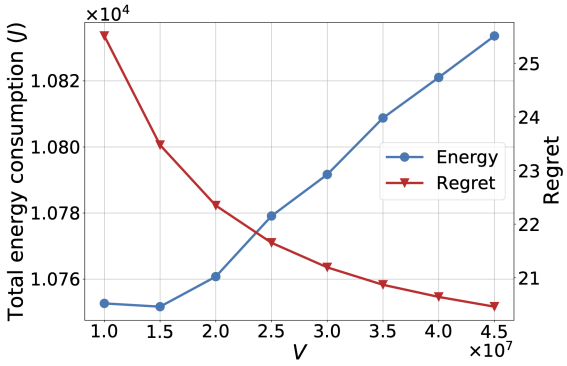


Fig. 6. Tradeoff between regret and energy consumption.

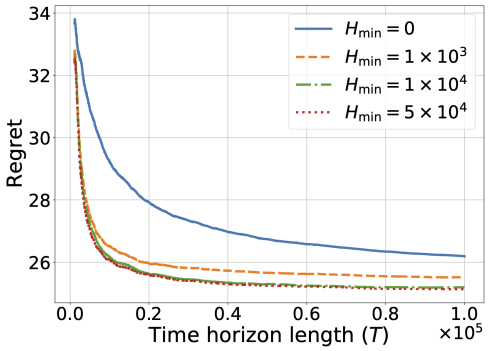


Fig. 7. Regret under fixed values of $H_{\min}$ when $V = 10^7$.

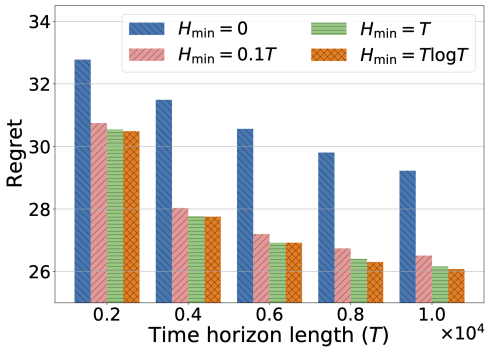


Fig. 8. Regret under different values of $H_{\min}$ when $V = 10^7$.

Figure 4(b)) converge to their energy budgets $\gamma_0 = 115$ J and $\gamma_{10} = 300$ J, respectively. In other words, the energy consumption constraints are satisfied when the value of T is sufficiently large. Moreover, the convergence rates of time-averaged energy consumption curves increase with the decrease of the value of V. Such results imply that under a larger value of V, a longer time horizon length T is needed to satisfy the time-averaged energy consumption constraints.

Figure 5 illustrates the time-averaged energy consumptions on all 36 GSs over $T = 10^5$ time slots. When $V = 1.5 \times 10^7$, none of the GSs consumes more than 298 J of energy. As the value of V grows to 4.5×10^7, the energy consumptions of more than half of the GSs are greater than 298 J. The results imply that the energy consumptions on GSs increase with the increase of the value of V. Nonetheless, none of the GSs consumes more than 300 J of energy, *i.e.*, the time-averaged energy consumption constraints in (10) and (12) are satisfied under different values of V.

6.2.2 Tradeoff between Regret and Total Energy Consumption

Next, we investigate the tradeoff between regret and total time-averaged energy consumption incurred in the network. As shown in Figure 6, with a larger value of V, the value of regret reduces while the total energy consumption increases. Such results illustrate a tunable tradeoff between regret and total energy consumption, which verifies our theoretical analysis in Section 5.

6.2.3 Regret under Different Values of Time Horizon Length and Numbers of Offline Historical Observations

In Figure 7, we illustrate how the regret of TRDBL evolves as the time horizon length T increases from 1.1×10^3 to 10^5 when the numbers $H_{\min}$ of offline historical observations are fixed as 0, 10^3, 10^4, and 5×10^4, respectively. Note that when $H_{\min} = 0$, there is no available offline historical observation. The figure shows that under a fixed number of offline historical observations, the regret of TRDBL decreases as the time horizon length T increases. Besides, when the value of T is fixed, a lower regret is achieved given a larger number of offline historical observations. Such a regret reduction becomes less obvious as the value of T increases. For example, as the value of $H_{\min}$ increases from 0 to 5×10^4, the regret reduces by 10.43% when $T = 10^4$, and only by 4.05% when $T = 10^5$.

Next, we investigate the regret of TRDBL under different values of time horizon length T and different numbers $H_{\min}$ of offline historical observations. In Figure 8, we consider the cases when $T \in \{2 \times 10^3, 4 \times 10^3, 6 \times 10^3, 8 \times 10^3, 10^4\}$. Given each value of T, we set the value of $H_{\min}$ to be 0, $0.1T$, T, $T \log T$, respectively. From the figure we see that under a small value of $H_{\min}$, the benefit brought by increasing the number of offline historical observations is noticeable. But given a large value of $H_{\min}$, such a reduction becomes less obvious. For example, given $T = 10^4$, the regret reduces by 9.26% as $H_{\min}$ increases from 0 to $0.1T$ (*i.e.*, increased by 10^3), and only reduces by 1.33% as $H_{\min}$ increases from $0.1T$ to T (*i.e.*, increased by 9×10^3).

In summary, given more offline historical information, the learning efficiency of TRDBL can be improved. However, the improvement becomes less significant as the value of $H_{\min}$ increases. Such results are consistent with our theoretical analysis in Section 5.2.

6.2.4 Performance under Different Task Arrival Rates

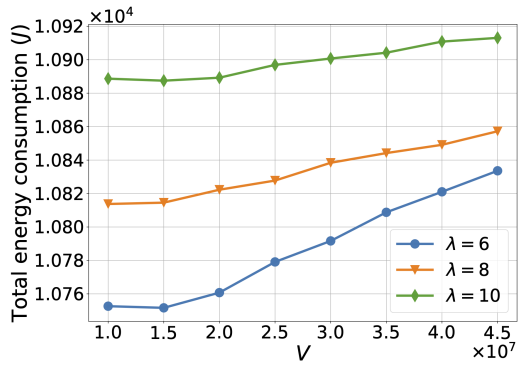
We investigate the performance of TRDBL under different values of task arrival rate λ in Figure 9. From Figure 9(a) we see that given $V = 1.5 \times 10^7$, the total energy consumption increases by 1.95% as the value of λ increases from 6 to 10. The reason is that as more tasks arrive, more energy would be consumed for data transmission and task processing. Under the same settings, Figures 9(b) and 9(c) show that the time-averaged total task latency and the value of regret increase by 82.40% and 22.94%, respectively. This is because given more task arrivals, the LEO satellite would transmit more tasks to non-optimal GSs to satisfy the energy constraints with respect to optimal GSs, resulting in a longer total task latency and a higher regret. Such results verify our theoretical analysis in Theorem 2 (see Section 5.2).

6.2.5 Regret under Different Exploration Strategies

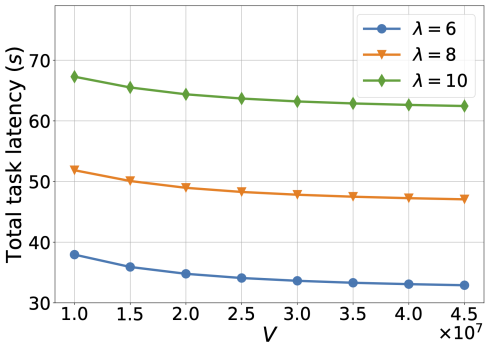
As introduced in Section 4.2, the confidence radius in the estimation (17) incentivizes TRDBL to explore the transmission latencies from the LEO satellite to those GSs with most uncertain estimates. Particularly, the larger the confidence radius, the higher the probability for exploration. To investigate the regret under different exploration strategies, we propose two variants of TRDBL by employing UCB1-Tuned (UCBT) [35] and ε-greedy [33] algorithms in the algorithm design, respectively. Details of the two variants are shown as follows.

- *TRDBL-UCBT*: As a variant of TRDBL, TRDBL-UCBT replaces the estimate $\tilde{w}_m(t)$ in line 9 of Algorithm 1 with the UCBT estimate of w_m for all $m \in \mathcal{M}$, and the rest remains the same as TRDBL.
- *TRDBL-greedy*: TRDBL-greedy employs the idea of ε-greedy algorithm to explore with a probability of ε in the decision-making process of task scheduling ($\varepsilon \geq 0$). Specifically, TRDBL-greedy differs from TRDBL in lines 9-10 of Algorithm 1. It replaces the estimate $\tilde{w}_m(t)$ in line 9 with the empirical mean $\bar{w}_m(t)$ for all $m \in \mathcal{M}$. Then in each time slot t, with probability ε, for each task, its assigned GS is uniformly randomly selected from the available GS set $\mathcal{M}(t)$. With probability $1 - \varepsilon$, the GS is uniformly randomly selected from the candidate GS set $\mathcal{M}_{\min}(t)$ obtained in line 9. The rest of TRDBL-greedy remains the same as TRDBL.

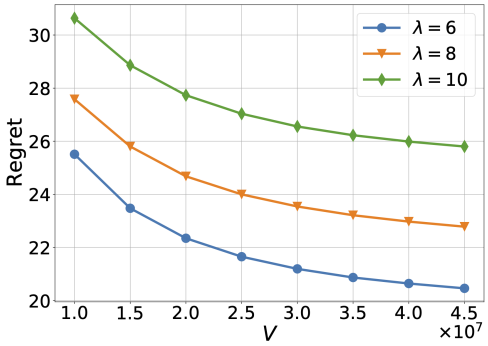
We compare the performance of TRDBL, TRDBL-UCBT, and TRDBL-greedy ($\varepsilon \in \{0, 0.01, 0.2\}$) in Figure 10, where the asterisk ("$*$") stands for "TRDBL". The results show that when the value of V increases, all schemes have the same trends of regret change. It is noteworthy that though TRDBL-greedy with $\varepsilon = 0$ has no explicit chance of uniform exploration, it still performs as well as TRDBL, TRDBL-UCBT, and TRDBL-greedy with $\varepsilon = 0.01$. The reason is that enforced exploration is conducted in the task scheduling



(a) Total energy consumption.



(b) Total task latency.



(c) Regret.

Fig. 9. Performance of TRDBL under different values of λ.

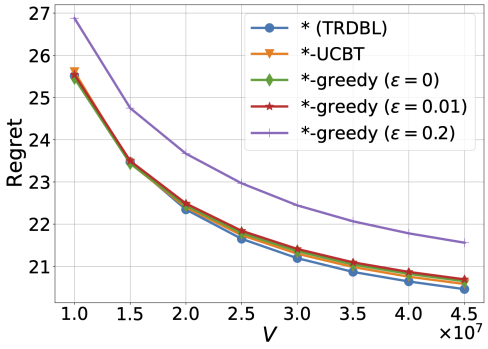


Fig. 10. Comparison of TRDBL and its variants.

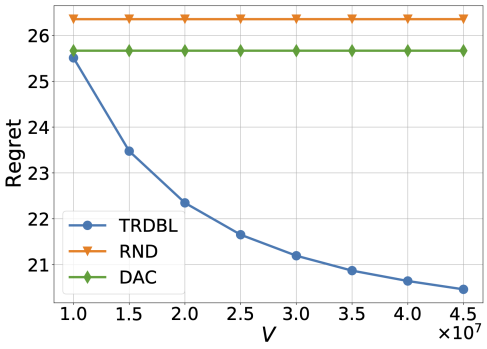
process to satisfy the energy consumption constraints. Besides, among all schemes, TRDBL-greedy with $\varepsilon = 0.2$ has the worst regret performance due to its over-exploration in the task scheduling process. Last but not least, the results also indicate that the performance of TRDBL-greedy is subject to the value choice of ε, while both TRDBL and TRDBL-UCBT can free from the laborious parameter tuning.

6.2.6 TRDBL vs. Baseline Schemes

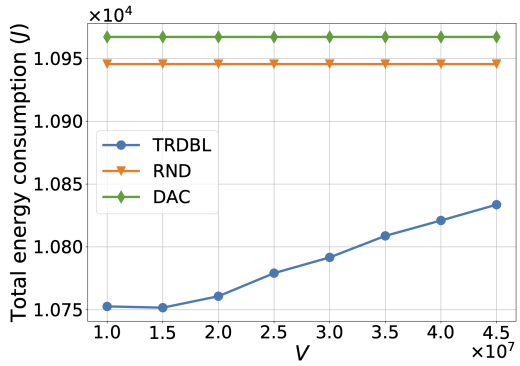
We compare the performance of TRDBL with two baseline schemes: RND (Random) and DAC (Deep Actor Critic) [9]. The details about how the two baseline schemes proceed are shown as follows.

- RND: Under RND, the LEO satellite schedules each task to a GS which is randomly and uniformly selected from its accessible GS set.
- DAC: DAC is a state-of-the-art scheme which is developed based on deep reinforcement learning techniques. The key idea of DAC is to employ the actor-critic method to learn the optimal decisions under uncertain environment dynamics. Specifically, under DAC, the LEO satellite maintains two neural networks: an actor network and a critic network. In each time slot, the actor network takes the observed environment dynamics as input and generates scheduling decisions for tasks. The critic network takes both the environment dynamics and task scheduling decisions as input and evaluates the decisions with respect to task latencies. In an iterative manner, the LEO satellite updates the critic network to improve the accuracy of the evaluation, and updates the actor network in the direction suggested by the evaluation to improve its policy and achieve a lower regret.

The performance comparison of TRDBL against the two baseline schemes is shown in Figure 11. Note that the total energy consumptions and regrets under RND and DAC remain unchanged under different values of V. The reason is that the parameter V is not involved in the decision making of these schemes. The results show that TRDBL achieves the lowest regret value and the smallest time-averaged energy consumption among the three schemes. For example, in Figure 11(a), given $V = 3 \times 10^7$, the regrets under RND and DAC are 24.35% and 21.11% higher than that under TRDBL, respectively. In Figure 11(b), given $V = 3 \times 10^7$, the total energy consumptions under RND and DAC are 1.43% and 1.63% more than that under TRDBL, respectively. Moreover, in contrast to TRDBL, the other two schemes (RND and DAC) fail to satisfy all the time-averaged energy constraints in (10) and (12).[3] In summary, TRDBL achieves a better performance than the two baseline schemes in terms of both regret value and energy consumptions.



(a) Regret.



(b) Total energy consumption.

Fig. 11. Comparison of TRDBL and baseline schemes.

7 Conclusion

In this article, we considered the joint problem of task scheduling and resource allocation with unknown wireless channel conditions between space and ground in satellite-terrestrial integrated networks with the goal of minimizing task latencies. We formulated the problem as a constrained CMAB problem and designed a novel task scheduling and resource allocation scheme called TRDBL by integrating online control, online learning, and offline historical information. Our theoretical analysis and simulation results showed that TRDBL achieves a sublinear time-averaged regret under energy consumption constraints.

3. Recall that the energy budgets for the LEO satellite and each of the 36 GSs are set as 115 J and 300 J, respectively. Therefore, the total time-averaged energy consumption of the LEO satellite and all GSs should not exceed 10915 J. However, the time-averaged total energy consumptions under RND and DAC are 10945.48 J and 10967.07 J, respectively.

References

[1] J. Wang, X. Gao, X. Huang, Q. Leng, Z. Shao, and Y. Yang, "Energy-constrained online matching for satellite-terrestrial integrated networks," in *Proceedings of IEEE ICC*, 2021.

[2] B. Di, L. Song, Y. Li, and H. V. Poor, "Ultra-dense leo: Integration of satellite access networks into 5g and beyond," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 62–69, 2019.

[3] X. Jia, T. Lv, F. He, and H. Huang, "Collaborative data downloading by using inter-satellite links in leo satellite networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1523–1532, 2017.

[4] Y. Wang, M. Sheng, W. Zhuang, S. Zhang, N. Zhang, R. Liu, and J. Li, "Multi-resource coordinate scheduling for earth observation in space information networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 268–279, 2018.

[5] 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, 2019.

[6] C. Niephaus, M. Kretschmer, and G. Ghinea, "Qos provisioning in converged satellite and terrestrial networks: A survey of the state-of-the-art," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2415–2441, 2016.

[7] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilen, R. C. Reinhart, and D. J. Mortensen, "Reinforcement learning for satellite communications: from leo to deep space operations," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 70–75, 2019.

[8] X. Li, W. Feng, Y. Chen, C.-X. Wang, and N. Ge, "Maritime coverage enhancement using uavs coordinated with hybrid satellite-terrestrial networks," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2355–2369, 2020.

[9] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for iot applications: A learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.

[10] J. Li, K. Xue, D. S. Wei, J. Liu, and Y. Zhang, "Energy efficiency and traffic offloading optimization in integrated satellite/terrestrial radio access networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2367–2381, 2020.

[11] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proceedings of ICML*, 2013.

[12] M. Zhang and W. Zhou, "Energy-efficient collaborative data downloading by using inter-satellite offloading," in *Proceedings of IEEE GLOBECOM*, 2019.

[13] Y. Wang, M. Sheng, J. Li, X. Wang, R. Liu, and D. Zhou, "Dynamic contact plan design in broadband satellite networks with varying contact capacity," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2410–2413, 2016.

[14] H. Huang, S. Guo, W. Liang, K. Wang, and A. Y. Zomaya, "Green data-collection from geo-distributed iot networks through low-earth-orbit satellites," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 806–816, 2019.

[15] H. Huang, S. Guo, W. Liang, K. Wang, and Y. Okabe, "Coflow-like online data acquisition from low-earth-orbit datacenters," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2743–2760, 2019.

[16] L. He, J. Li, M. Sheng, R. Liu, K. Guo, and D. Zhou, "Dynamic scheduling of hybrid tasks with time windows in data relay satellite networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4989–5004, 2019.

[17] P. Shivaswamy and T. Joachims, "Multi-armed bandit problems with history," in *Proceedings of AISTATS*, 2012.

[18] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[19] A. Alsharoa and M.-S. Alouini, "Improvement of the global connectivity using integrated satellite-airborne-terrestrial networks with resource optimization," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5088–5100, 2020.

[20] Z. Yang, Y. Li, P. Yuan, and Q. Zhang, "Tcsc: A novel file distribution strategy in integrated leo satellite-terrestrial networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5426–5441, 2020.

[21] J. Du, C. Jiang, Q. Guo, M. Guizani, and Y. Ren, "Cooperative earth observation through complex space information networks," *IEEE Wireless Communications*, vol. 23, no. 2, pp. 136–144, 2016.

[22] D. Lopez-Perez, A. Valcarce, G. De La Roche, and J. Zhang, "Ofdma femtocells: A roadmap on interference avoidance," *IEEE Communications Magazine*, vol. 47, no. 9, pp. 41–48, 2009.

[23] S. Zhang, D. Zhu, and Y. Wang, "A survey on space-aerial-terrestrial integrated 5g networks," *Computer Networks*, vol. 174, p. 107212, 2020.

[24] O. Somekh, O. Simeone, Y. Bar-Ness, A. M. Haimovich, and S. Shamai, "Cooperative multicell zero-forcing beamforming in cellular downlink channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3206–3219, 2009.

[25] D. Vasisht, J. Shenoy, and R. Chandra, "L2d2: low latency distributed downlink for leo satellites," in *Proceedings of ACM SIGCOMM*, 2021.

[26] S.-g. Kim, H. Eom, H. Y. Yeom, and S. L. Min, "Energy-centric dvfs controlling method for multi-core platforms," *Computing*, vol. 96, no. 12, pp. 1163–1177, 2014.

[27] A. A. Bisu, A. Purvis, K. Brigham, and H. Sun, "A framework for end-to-end latency measurements in a satellite network environment," in *Proceedings of IEEE ICC*, 2018.

[28] D. Zhou, M. Sheng, B. Li, J. Li, and Z. Han, "Distributionally robust planning for data delivery in distributed satellite cluster network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3642–3657, 2019.

[29] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[30] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[31] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," in *Proceedings of IEEE INFOCOM*, 2019.

[32] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[33] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *Proceedings of ECML*, 2005.

[34] N. Budhdev, M. C. Chan, and T. Mitra, "Pr^3: Power efficient and low latency baseband processing for lte femtocells," in *Proceedings of IEEE INFOCOM*, 2018.

[35] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002.

Xin Gao (Graduate Student Member, IEEE) received the B.Eng. degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, in 2015. She is currently pursuing the Ph.D. degree with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China, also with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China, and also with the University of Chinese Academy of Sciences, Beijing, China. Her current research interests include: intelligent networks, bandit and reinforcement learning, edge computing and Internet of Things.

Jingye Wang received his B.Eng. degree from the School of Computer Science and Engineering, South China University of Technology, Guangdong, China. He is currently pursuing his Master degree at the School of Information Science and Technology at ShanghaiTech University, Shanghai, China. His current research interest lies in reinforcement learning.

Xi Huang (Member, IEEE) received the B.Eng. degree from Nanjing University, China, in 2014, and the PhD degree from ShanghaiTech University, China, in 2021. He is currently with Shenzhen Institute of Artificial Intelligence and Robotics for society (AIRS). He was a visiting student at the Department of Electrical Engineering and Computer Sciences at UC Berkeley in 2017. His current research interests include the optimization for Intelligence networks and multi-agent learning.

Qiuyu Leng received her B.Eng. degree from Northeastern University, China, in 2019. She is currently pursuing the Master degree at School of Information Science and Technology, ShanghaiTech University, Shanghai, China. Her current research interests include: bandit and reinforcement learning, edge computing and Internet of Things.

Ziyu Shao (Senior Member, IEEE) received the B.S. and M.Eng. degrees from Peking University in 2001 and 2004, respectively, and the Ph.D. degree from the Chinese University of Hong Kong in 2010. He then worked as a Postdoctoral Researcher with the Chinese University of Hong Kong from 2011 to 2013. He is currently an Associate Professor with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. He was a Visiting Postdoctoral Researcher with the EE Department, Princeton University in 2012. He was also a Visiting Professor with the EECS Department, UC Berkeley in 2017. His current research interests center on intelligent networks, including AI for networks and networks for AI.

Yang Yang (Fellow, IEEE) received the B.S. and M.S. degrees in Radio Engineering from Southeast University, Nanjing, China, in 1996 and 1999, respectively, and the PhD degree in Information Engineering from the Chinese University of Hong Kong in 2002. He is currently a full professor with School of Information Science and Technology, the Master of Kedao College, as well as the Director of Shanghai Institute of Fog Computing Technology (SHIFT), ShanghaiTech University, China. He is also an Adjunct Professor with the Research Center for Network Communication, Peng Cheng Laboratory, China, as well as a Senior Consultant for Shenzhen SmartCity Technology Development Group, China. Before joining ShanghaiTech University, he has held faculty positions at the Chinese University of Hong Kong, Brunel University, U.K., University College London (UCL), U.K., and SIMIT, CAS, China. Yang's research interests include fog computing networks, service-oriented collaborative intelligence, wireless sensor networks, IoT applications, and advanced testbeds and experiments. He has published more than 300 papers and filed more than 80 technical patents in these research areas. He has been the Chair of the Steering Committee of Asia-Pacific Conference on Communications (APCC) since January 2019.