# Adapting to Dynamic LEO-B5G Systems: Meta-Critic Learning Based Efficient Resource Scheduling

Yaxiong Yuan, *Student Member, IEEE*, Lei Lei, *Member, IEEE*, Thang X. Vu, *Member, IEEE*, Zheng Chang, *Senior Member, IEEE*, Symeon Chatzinotas, *Senior Member, IEEE*, and Sumei Sun, *Fellow, IEEE*

***Abstract*—Low earth orbit (LEO) satellite-assisted communications have been considered as one of the key elements in beyond 5G systems to provide wide coverage and cost-efficient data services. Such dynamic space-terrestrial topologies impose an exponential increase in the degrees of freedom in network management. In this paper, we address two practical issues for an over-loaded LEO-terrestrial system. The first challenge is how to efficiently schedule resources to serve a massive number of connected users, such that more data and users can be delivered/served. The second challenge is how to make the algorithmic solution more resilient in adapting to dynamic wireless environments. We first propose an iterative suboptimal algorithm to provide an offline benchmark. To adapt to unforeseen variations, we propose an enhanced meta-critic learning algorithm (EMCL), where a hybrid neural network for parameterization and the Wolpertinger policy for action mapping are designed in EMCL. The results demonstrate EMCL's effectiveness and fast-response capabilities in over-loaded systems and in adapting to dynamic environments compare to previous actor-critic and meta-learning methods.**

***Index Terms*—LEO satellites, resource scheduling, reinforcement learning, meta-critic learning, dynamic environment.**

## I. Introduction

In beyond 5G networks (B5G), the massive number of connected users and their increasing demands for high-data-rate services can lead to overloading of terrestrial base stations (BSs), which in turn results in degraded user experience, e.g., longer delay in requesting data services or lower data rate [1]. In order to improve the network performance and user experience, the integration of satellites, e.g., low earth orbit (LEO) satellites, and terrestrial systems is considered as a promising solution to provide cost-efficient data services [2]. The solutions for terrestrial network optimization and resource management might not be suitable for direct application to integrated satellite-terrestrial systems [3]. In the literature, tailored schemes have been investigated to improve the networks' performance. In [4], the authors proposed a user scheduling scheme to maximize the sum-rate and the number of accessed users by utilizing the LEO-based backhaul. In [5], a joint power allocation and user scheduling scheme was proposed to maximize the network throughput in hierarchical LEO systems with the constraint of transmission delay. In [6], the authors developed a joint resource block allocation and power allocation algorithm to maximize the total transmission rate for LEO systems. It is worth noting that the resource optimization problems in LEO-terrestrial networks are typically combinatorial and non-convex. The conventional iterative optimization methods, e.g., in [4]–[6], are unaffordable for real-time operations due to their high computational complexity.

### A. Related Works: State-of-the-art and Limitations

Towards an efficient solution, various learning techniques have been studied. Compared to supervised learning, reinforcement learning (RL) learns the optimal policy from observed samples without preparing labeled data. As one of the promising RL methods, deep reinforcement learning (DRL) adopts deep neural networks (DNNs) for parameterization and rapid decision making. Recent works have applied RL/DRL for resource management in LEO-terrestrial systems [7]–[9]. In [7], to maximize the achievable rate in LEO-assisted relay networks, a DQN-based algorithm was proposed to make the online decisions for link association. The authors in [8] adopted multi-agent reinforcement learning to minimize the average number of handovers and improve the efficiency of channel utilization for LEO satellite systems. In [9], the authors applied an actor-critic (AC) algorithm to LEO resource allocation, such as beam allocation and power control. The above RL algorithms in practical LEO systems are limited by the following issue. That is, the performance of a learning model largely depends on the data originated from the experienced samples or the observed environment, but the wireless environment is highly complex and dynamic. When network parameters vary dramatically, the performance of the learning models can be degraded. To remedy this, one has to re-collect a large number of training data and re-train the learning models, which is time-consuming and inefficient to adapt to fast variations [10].

The work has been supported by the ERC project AGNOSTIC (742648), by the FNR CORE projects ROSETTA (C17/IS/11632107), FlexSAT (C19/IS/13696663), SmartSpace (C21/IS/16193290), and by the FNR bilateral project LARGOS (12173206). (Corresponding author: Lei Lei)

Yaxiong Yuan, Thang X. Vu, and Symeon Chatzinotas are with the Interdisciplinary Centre for Security, Reliability and Trust, Luxembourg University, 1855 Kirchberg, Luxembourg (e-mail: yaxiong.yuan@uni.lu; thang.vu@uni.lu; symeon.chatzinotas@uni.lu).

Lei Lei is with the School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: lei.lei@xjtu.edu.cn).

Zheng Chang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China, and also with the Faculty of Information Technology, University of Jyväskylä, FI-40014 Jyväskylä, Finland (e-mail: zheng.chang@jyu.fi).

S. Sun is with the Institute for Infocomm Research, Agency for Science, Technology, and Research, Singapore 138632 (e-mail: sunsm@i2r.a-star.edu.sg).

To address this issue, a variety of studies focus on how to make the learning models quickly respond to dynamic environments. Transfer learning applies the knowledge acquired from a source learning task to a target learning task to speed up the re-training process and reduce the volume of the collected new data sets [11]. The performance of transfer learning is limited by finding correlated tasks. Another approach, joint learning, aims at obtaining a single model that can be adapted to dynamic environments by optimizing the loss function over multiple tasks [12]. Besides, continual learning can also accelerate the adaptation to the new learning task by adding the experienced data from the previous tasks to the re-training data set, thus avoiding completely forgetting previously learned models [13]. Joint learning and continual learning might have good learning performance on average but have limited generalization abilities when different tasks are highly diversified [14]. In contrast, meta-learning extracts meta-knowledge and achieves good performance for specific tasks without requiring the related source tasks. The authors in [15] proposed a model-agnostic meta-learning algorithm (MAML) to obtain the model's initial parameters as meta-knowledge to quickly adapt to new tasks. In [16], an algorithm combining actor-critic with MAML (AC-MAML) was developed to learn a new task from fewer experience data sets. In [17], the authors proposed a promising meta-critic learning framework with better performance than conventional AC and AC-MAML. In [18], a meta-learning-based adaptive sensing algorithm was proposed, which determines the next most informative sensing location in wireless sensor networks. In [19], meta-learning was applied to find a common initialization vector that enables fast training of an autoencoder for the fading channels. Most of the meta-learning methods were applied in the areas of pattern recognition [15], robotics [16], [17], and physical layer communications [19], which typically address simple learning tasks with limited action space. However, when the learning techniques, e.g., DRL, AC-MAML, or meta-critic learning, are applied to address combinatorial optimization problems in a dynamic LEO-terrestrial network, the action space can be huge and the input-output relationships can become more complex. These may degrade the efficiency of the above learning methods.

### B. Motivations and Contributions

Moving beyond the state-of-the-art, this paper intends to address the following questions:

- How to make the learning solutions more adaptive to dynamic LEO-terrestrial networks?
- How to deal with the huge action space and improve the learning efficiency?

In this study, we design an enhanced meta-critic learning algorithm (EMCL) to enable efficient resource scheduling for dynamic LEO-terrestrial systems, and emphasize the solutions to deal with non-ideal dynamic environments. The major contributions are summarized as follows:

- We design a tailored metric for over-loaded LEO systems with dense user distribution, aiming at serving more users and delivering a higher volume of requested data.
- We formulate the resource scheduling problem as a quadratic integer programming (QIP) and provide two offline optimization-based benchmarks, i.e., optimal branch and bound (B&B) algorithm and suboptimal alternating direction method of multipliers-based heuristic algorithm (ADMM-HEU).
- Due to the combinatorial nature and the high complexity of the offline solutions, we solve the problem from the perspective of DRL by reformulating a Markov decision process (MDP) to make online decisions with the identical objective as the original problem.
- To adapt to dynamic environments, we propose an EMCL algorithm based on a meta-critic framework. Compared to conventional meta learning, the novelty stems from that: 1) The critic has good generalization abilities to evaluate any new task such that the learning agent can adjust the policy timely when the environment changes; 2) The tailored design of a hybrid neural network extracts the features from the current and historical samples; 3) the integrated Wolpertinger policy allows the actor to make decisions more efficiently in an exponentially increasing action space.
- We evaluate the proposed EMCL with other benchmarks in three practical dynamic scenarios, i.e., bursty user demands, dramatically fluctuated channel states, and user departure/arrival. The numerical results verify EMCL's effectiveness and fast-response capabilities in adapting to dynamic environments.

The rest of the paper is organized as follows. The system model is presented in Section II. We formulate a resource scheduling problem and develop optimal and suboptimal solutions for performance benchmarks in Section III. In Section IV, we model the problem as an MDP and develop an EMCL algorithm. Numerical results are demonstrated and analyzed in Section V. Finally, Section VI concludes the paper.

## II. System Model and Problem Formulation

### A. LEO-Terrestrial Network

In practice, terrestrial BSs can become over-loaded and congested. This common issue has received considerable attention from academia, industry, and standardization bodies, e.g., 3GPP Release 17 [20]. In this work, we address this challenging issue via developing satellite-aided solutions. As shown in Fig. 1, the BSs with limited resources might not be able to serve all the users and deliver all the requested data demands within a required transmission or queuing delay. To relieve the burden of the terrestrial BSs, LEO satellites are introduced to offload traffic from BSs or provide backhauling services. The LEO employs a transparent payload. For spectrum usage, the system keeps consistent with currently deployed space and ground systems. That is, the LEO satellites operate at the Ka-band to provide broadband services to advanced terminals, e.g., equipped with very small aperture terminals (VSAT), while the 5G terrestrial system adopts sub-6GHz at the C-band to serve normal mobile devices, e.g., smartphones [21].

We consider two types of mobile terminals (MTs) in the system. The first type is the normal cellular terminals, e.g.,

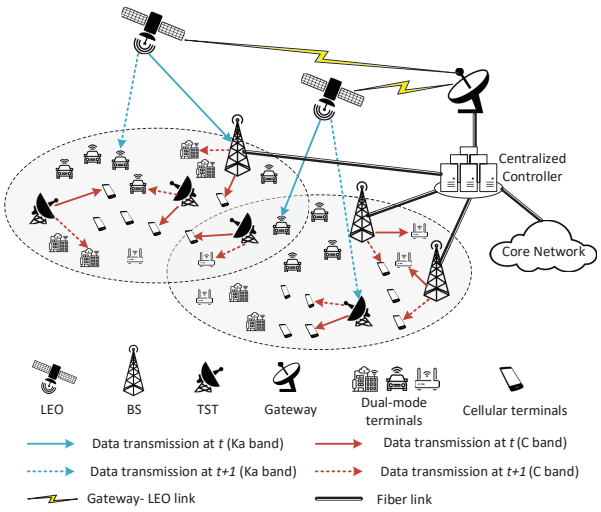



Fig. 1. An illustrative LEO-terrestrial communication system

cell phones, that can be served by BSs or terrestrial-satellite terminals (TSTs), but cannot be served by LEO due to the size limitation of dish antennas. The other is the dual-mode terminals, e.g., vehicular terminals, which are equipped with a 3GPP terrestrial-non-terrestrial network (TN-NTN) compliant dual-mode that can be either served by LEO via Ka-band (in rural areas) or by BS/TST through C-band (in urban areas) [22]. Compared to conventional cellular BS, TST is a small-size terminal that acts as a flexible and cost-saving access point, e.g., Starlink ground terminals. A TST can receive backhauling services from LEO over Ka-band and transmit data to MTs over C-band [4]. The terrestrial BSs can request data from the core network through optical fiber links or from the LEO satellites through the BS-LEO link. We remark that Fig. 1 can be extended to a large-scale network with a massive number of MTs. Specifically, an MT in Fig. 1 can represent a cluster of densely-deployed devices. Due to the proximity, the channel states of the devices within a cluster can be assumed identical. When a cluster is scheduled, all the devices within the cluster will be scheduled by the TDMA (or FDMA) mode to avoid intra-cluster interference.

We denote $\mathcal{S}, \mathcal{B}, \mathcal{M}$ and $\mathcal{L}$ as the set of TSTs, BEs, MTs, and LEOs, respectively, where $\mathcal{M}$ is the union of set $\mathcal{M}_1$ (all the cellphone MTs) and $\mathcal{M}_2$ (all the dual-mode MTs). Thus, the union of receivers, i.e., ground devices (GDs), can be expressed as $\mathcal{K} = \mathcal{S} \cup \mathcal{B} \cup \mathcal{M} = \{1, ..., k, ..., K\}$, where $K = |\mathcal{S}| + |\mathcal{B}| + |\mathcal{M}|$. Similarly, the union of transmitters is written by $\mathcal{N} = \mathcal{S} \cup \mathcal{B} \cup \mathcal{L} = \{1, ..., n, ..., N\}$, where $N = |\mathcal{S}| + |\mathcal{B}| + |\mathcal{L}|$. The time domain is divided by time slots, i.e., $\mathcal{T} = \{1, ..., t, ..., T\}$. In data transmission, each transmitter $n$ serves a GD in unicast mode, i.e., no joint transmission and no multi-cast transmission. Within a time slot, multiple transmitter-GD links can be activated, forming a link group. We denote $\mathcal{G} = \{1, ..., g, ..., G\}$ as a set by enumerating all the valid link groups.

To coordinate the link scheduling between terrestrial and satellite parts, a centralized controller is deployed in the system [23]. With the centralized controller, the information from the ground and satellite can be collected and exchanged, which facilitates the implementation of scheduling decisions. In addition, efficient synchronization approaches can be implemented on the transmitters and receivers to guarantee that the resource scheduling updates are performed accurately in LEO satellite systems [24].

### B. Channel Modeling

We consider time-varying channels for both satellite and terrestrial communication. At time slot $t$, the channel state between receiver $k$ and transmitter $n$ can be modeled as:

$$h_{k,n,t} = \begin{cases} G_{leo}^{(T)} \cdot G_{k,n,t}^{(C)} \cdot G^{(R)}, & n \in \mathcal{L}, \\ G_{ter}^{(T)} \cdot G_{k,n,t}^{(C)} \cdot G^{(R)}, & n \in \mathcal{N} \backslash \mathcal{L}, \end{cases} \tag{1}$$

where $G_{leo}^{(T)}$ and $G_{ter}^{(T)}$ are the transmit antenna gain of LEO and terrestrial BS/TST, respectively. We assume that all the GDs are equipped with a single receiving antenna, so that their receive antenna gains $G^{(R)}$ are uniform. $G_{k,n,t}^{(C)}$ represents the channel fading between transmitter $n$ and GD $k$ at time slot $t$. For LEO-to-GD channel, a widely used channel fading model in [4], [6], [25] is adopted, which includes free-space path loss, pitch angle fading, atmosphere fading, and Rician small-scale fading:

$$G_{k,n,t}^{(C)} = \left( \frac{c}{4\pi d_{k,n,t} f_{leo}} \right)^2 \cdot G_{k,n}^{(P)} \cdot A(\Omega) \cdot \varphi, \tag{2}$$

where $c$ is the speed of light, $d_{k,n,t}$ is the propagation distance between LEO and the terminals, $f_{leo}$ is the carrier frequency of LEO, $G_{k,n}^{(P)}$ is the pitch angle fading gain, and $\varphi$ is the Rician fading gain. The atmospheric fading gain $A(\Omega)$ is the function of the angle $\Omega$, where $\sin \Omega = H / d_{k,n,t}$, and $H$ is the altitude of LEO.

$$A(\Omega) = 10^{\left(\frac{3\chi}{10 \sin \Omega}\right)}, \tag{3}$$

where $\chi$, in $dB/km$, is the attenuation through the clouds and rain. In downlink transmission, we assume that Doppler shift caused by the high mobility of LEO can be perfectly pre(post)-compensated in the gateway based on the predictable satellite motion and speed [26]. For terrestrial channels, i.e., TST/BS-to-MT, $G_{k,n,t}^{(C)}$ consists of the path loss and Rayleigh small-scale fading [27], which is given by:

$$G_{k,n,t}^{(C)} = \left( \frac{c}{4\pi d_{k,n,t} f_{ter}} \right)^2 \cdot \phi, \tag{4}$$

where $f_{ter}$ is the carrier frequency of TST/BS and $\phi$ is the Rayleigh fading factor.

Based on the adopted channel fading models (2) and (4), we further model the time-varying channel as the finite state Markov channel (FSMC) to capture the time-correlation characteristics and conduct mathematically tractable analysis. To form an FSMC, we first discretize the channel state $h_{k,n,t}$ into $L$ levels, i.e., $\mathcal{H} = \{h_1, ..., h_L\}$, where the thresholds $h_l, ..., h_L$ are determined by the equal-probability method [28]. Then the transition probability matrix is defined as:

$$\mathbf{P} = \begin{bmatrix} P_{1,1} & \cdots & P_{1,L} \\ \vdots & \ddots & \vdots \\ P_{L,1} & \cdots & P_{L,L} \end{bmatrix}, \tag{5}$$

where the transition probability $P_{l,l'}$ can be written as:

$$P_{l,l'} = \text{Prob}\left[h_{k,n,t+1}{=}h_{l'}|h_{k,n,t}{=}h_l\right],\ h_l, h_{l'} \in \mathcal{H}. \quad (6)$$

That is, at a given time slot $t$, if $h_{k,n,t} = h_l$, $P_{l,l'}$ refers to the probability of channel state at the next time slot $h_{k,n,t+1}$ transiting from $h_l$ to $h_{l'}$, which can be approximated by the ratio between the level crossing rate and the average number of symbol per second [28].

### C. Optimization Problem

We formulate a resource scheduling problem for the considered over-loaded LEO-5G systems. We use binary indicators $\alpha_{k,n,g}$ to represent the activated links in group $g \in \mathcal{G}$, where $\alpha_{k,n,g} = 1$ if the transmitter-GD link $(n,k)$ is included in group $g$ and will be activated when group $g$ is scheduled, otherwise, 0. Set $\mathcal{G}$ and indicators $\alpha_{k,n,g}$ are the necessary input parameters for the optimization problem P1. Following the principles in (7)-(10), we enumerate valid links and candidate groups. In implementation, every enumerated link or group will undergo a feasiblity-check step to ensure that no links or groups violate (7)-(10).

$$\alpha_{k,n,g} = 0,\ \forall k \in \mathcal{K}\backslash\mathcal{M}, \forall n \in \mathcal{N}\backslash\mathcal{L}, \forall g \in \mathcal{G}, \quad (7)$$

$$\alpha_{k,n,g} = 0,\ \forall k \in \mathcal{M}_1,\ \forall n \in \mathcal{L},\ \forall g \in \mathcal{G}, \quad (8)$$

$$\sum\nolimits_{n\in\mathcal{N}} \alpha_{k,n,g} \leq 1,\ \forall k \in \mathcal{K}, \forall g \in \mathcal{G}, \quad (9)$$

$$\sum\nolimits_{k\in\mathcal{K}} \alpha_{k,n,g} \leq 1,\ \forall n \in \mathcal{N}, \forall g \in \mathcal{G}. \quad (10)$$

(7) and (8) exclude certain types of links, i.e., BS-BS, TST-TST, BS-TST, TST-BS, and LEO-cellphone. (9) means that each GD $k$ in group $g$ receives data from at most one transmitter, and (10) represents each transmitter $n$ in group $g$ serves no more than one GD. For example, consider a simple system with 1 LEO, 1 TST, 1 BS, and 2 MTs (an MT1 in $\mathcal{M}_1$, and an MT2 in $\mathcal{M}_2$). There are four possible receivers, i.e., TST, BS, MT1, and MT2, indexed by $K = \{1,2,3,4\}$, respectively, and three possible transmitters, i.e., TST, BS, and LEO, indexed by $\mathcal{N} = \{1,2,3\}$. Filtered by (7)-(8), all the valid links are $(1,3)$ (TST to MT1), $(1,4)$ (TST to MT2), $(2,3)$ (BS to MT1), $(2,4)$ (BS to MT2), $(3,1)$ (LEO to TST), $(3,2)$ (LEO to BS), and $(3,4)$ (LEO to MT2). Confined by (9)-(10), a combination of the above links can be a valid group g, e.g, a group $\{(3,4),(1,3)\}$ contains two links. Enumerating all the valid groups forms set $\mathcal{G} = \{\{(1,3),(2,4)\},\{(1,3),(3,4)\},.....,\{(1,3),(2,4),(3,1)\}\}$, which is served as the input set for decision making. Note that filtered by constraints (7)-(10), a large number of invalid links and groups have been excluded. For even larger networks, we remark that a full enumeration of groups might be unaffordable in implementation. To deal with this issue, some heuristic enumeration approaches can be adopted in pre-process stage to reduce the complexity to an affordable level [29].

Confined by (7) and (8), the SINR and the volume of transmitted data of GD $k$ in group $g$ at time slot $t$ are expressed in (11) and (12), respectively.

$$\gamma_{k,g,t} = \frac{\sum_{n\in\mathcal{L}} h_{k,n,t}\alpha_{k,n,g}p_{k,g}}{\sum_{j\in\mathcal{K}\backslash k}\sum_{n\in\mathcal{L}} h_{j,n,t}\alpha_{j,n,g}p_{k,g} + \sigma^2} + \frac{\sum_{n\in\mathcal{N}\backslash\mathcal{L}} h_{k,n,t}\alpha_{k,n,g}p_{k,g}}{\sum_{j\in\mathcal{K}\backslash k}\sum_{n\in\mathcal{N}\backslash\mathcal{L}} h_{j,n,t}\alpha_{j,n,g}p_{k,g} + \sigma^2}, \quad (11)$$

and

$$R_{k,g,t} = \Phi B_{k,g}\log_2(1+\gamma_{k,g,t}), \quad (12)$$

where $p_{k,g}$ is the transmit power to GD $k$ in group $g$ and $\Phi$ is the duration of each time slot. We denote $B_{leo}$ and $B_{ter}$ are the fixed bandwidth for LEO and BS/TST, respectively, such that the used bandwidth $B_{k,g}$ for GD $k$ in group $g$ can be calculated by $B_{leo}\sum_{n\in\mathcal{L}}\alpha_{k,n,g} + B_{ter}\sum_{n\in\mathcal{N}\backslash\mathcal{L}}\alpha_{k,n,g}$. We define the decision variables as $\mathbf{x} = [x_{1,1},...,x_{g,t},...,x_{G,T}]$ where

$$x_{g,t} = \begin{cases} 1, & \text{if group } g \text{ is scheduled at time slot } t, \\ 0, & \text{otherwise.} \end{cases}$$

In a practical over-loaded scenario, not all the terminals can be timely served and their actual demands may not be fully delivered in time due to massive access requests competing for limited resources. Under this undesirable scenario, the optimization task may shift from "serving all the terminals and satisfying all the demands" to "serving as many terminals (and their demands) as possible". On this basis, we denote $D_k$ and $D'_k(<D_k)$ as the actual demand (in bits) and the threshold, respectively. In the objective design, we consider a composite utility function in (13), and define that GD $k$ is served, i.e., $f_k(\mathbf{x}) = 1$, when a threshold $D'_k$ is satisfied.

$$f_k(\mathbf{x}) = \mathbb{1}\left(\sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} R_{k,g,t}x_{g,t} - D'_k\right), \quad (13)$$

where $\mathbb{1}(\cdot)$ is an indicator function such that $\mathbb{1}(\beta) = \begin{cases} 1, & \text{if } \beta > 0 \\ 0, & \text{if } \beta \leq 0 \end{cases}$. We introduce a threshold $D'_k$ in (13) since in an over-loaded scenario with densely deployed users, the system may not be able to satisfy all the actual demand $D_k$ within one scheduling cycle. In implementation, we predefine $D'_k = \varepsilon D_k$, where $0 \leq \varepsilon \leq 1$. The value of $\varepsilon$ is selected from the middle segment of $[0,1]$ to avoid too high or low value, such that $D'_k$ has a considerable impact on the optimization results and the trade-off effect.

We convert the non-linear function $f_k(\mathbf{x})$ to a linear function by introducing auxiliary variables $\mathbf{y} = [y_1,...,y_k,...,y_K]$ and linear constrains (14d), where $y_k = f_k(\mathbf{x})$. The optimization problem is formulated as:

$$\mathbf{P1}: \min_{x_{g,t},y_k}\ f(\mathbf{x},\mathbf{y}) = \eta_0\left(\sum_{k\in\mathcal{K}} y_k - K\right)^2 + \sum_{k\in\mathcal{K}}\eta_k\left(\sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} R_{k,g,t}x_{g,t} - D_k\right)^2 \quad (14a)$$

$$s.t.\ \bar{\gamma}_k - \gamma_{k,g,t} \leq V\left(1 - x_{g,t}\sum_{n\in\mathcal{N}}\alpha_{k,n,g}\right),$$

$$\forall k \in \mathcal{K}, g \in \mathcal{G}, t \in \mathcal{T}, \tag{14b}$$

$$\sum_{g \in \mathcal{G}} x_{g,t} \leq 1, \ \forall t \in \mathcal{T}, \tag{14c}$$

$$D'_k y_k \leq \sum_{t \in \mathcal{T}} \sum_{g \in \mathcal{G}} R_{k,g,t} x_{g,t}, \ \forall k \in \mathcal{K}, \tag{14d}$$

$$x_{g,t} \in \{0,1\}, \ \forall g \in \mathcal{G}, \ t \in \mathcal{T}, \tag{14e}$$

$$y_k \in \{0,1\}, \ \forall \ k \in \mathcal{K}, \tag{14f}$$

where $\bar{\gamma}_k$ is the SINR threshold of GD $k$, $V$ is a positive sufficiently large value, and $\eta_0, ..., \eta_K$ are the weight factors. Considering the users' fairness and resource utilization in an over-loaded system, we design a tailored utility function (14a) consisting of two components. The first term encourages serving more users and meeting their minimum requirement $D'_k$ since satisfying low-traffic users are more likely to have rewards in the objective. The second term aims at minimizing the supply-demand gap such that the scheduler tends to serve the users with higher demand $D_k$ or higher weights $\eta_k$ $(k = 1, ..., K)$. The priority or importance of the two parts can be adjusted by pre-defined weight values according to different scenarios. For example, when a large number of delay-sensitive and low-traffic users enter the network, the scheduler may give more priority by increasing $\eta_0$ to serve this type of users as many as possible, while the delay-tolerate services with high data demand may have lower priority (with decreased $\eta_k$) in this scheduling cycle.

- The constraints (14b) represent the SINR requirement in practical satellite and 5G systems. If GD $k$ in group $g$ is scheduled at time slot $t$, i.e., $x_{g,t} \sum_{n \in \mathcal{N}} \alpha_{k,n,g} = 1$, the SINR of GD $k$ should be higher than the threshold $\bar{\gamma}_k$ to guarantee the link quality. This also implies that scheduling many links with strong co-channel interference may not be a wise option in the optimal solution. The setting of $\bar{\gamma}_k$ refers to the standard of DVB-S2X [31] and 3GPP Release 16 [32].
- The constraints (14c) represent no more than one group can be scheduled in a time slot.
- In constraints (14d), we define that if GD $k$ is served, i.e., $y_k = 1$, the received data should be larger than $D'_k$.

## III. Characterization on Solution Development

In this section, we propose an optimal method and a heuristic approach as the offline benchmarks for small-medium and large-scale instances, respectively. In addition, we outline conventional online-learning solutions and their limitations.

### A. The Proposed Optimal and Sub-optimal Solutions

Towards the optimum of $\mathbf{P1}$, we first identify the convexity of $\mathbf{P1}$ when the binary variables are relaxed.

**Lemma 1.** *The relaxation problem of* $\mathbf{P1}$ *is convex.*

*Proof.* See Appendix A. □

Based on Lemma 1, we conclude that $\mathbf{P1}$ is an integer convex optimization problem. The optimum can be obtained by B&B that solves a convex relaxation problem at each node, with the complexity $\mathcal{O}(2^{G \times T + K})$ [33]. Although the complexity increases exponentially, the B&B-based approach can provide a performance benchmark at least for small-medium instances.

To reduce the complexity in solving large-scale problems, we develop a suboptimal algorithm. We observe that $\mathbf{P1}$ has a variable-splitting structure, which motivates the development of ADMM based approaches [34]. The algorithm is summarized in Alg. 1, first solving the convex relaxation problem of $\mathbf{P1}$ based on ADMM (in lines 2-8), followed by a rounding operation (in lines 9-13). In ADMM, we divide the relaxed variables into $T+1$ blocks $\hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_T, \hat{\mathbf{y}}$, where $\hat{\mathbf{x}}_t = [\hat{x}_{1,t}, ..., \hat{x}_{G,t}]$, and introduce auxiliary variables $\mathbf{z} = [z_1, ..., z_K]$, where

$$z_k = D'_k \hat{y}_k - \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} R_{k,g,t} \hat{x}_{g,t}, \forall k \in \mathcal{K}. \tag{15}$$

The inequality constraints (14d) are replaced by:

$$z_k \leq 0, \ \ \forall k \in \mathcal{K}. \tag{16}$$

The augmented Lagrangian function is expressed as:

$$\begin{aligned} & L(\hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_T, \hat{\mathbf{y}}, \mathbf{z}, \boldsymbol{\lambda}) \\ =& f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) + \sum_{k \in \mathcal{K}} \lambda_k \left( z_k - D'_k \hat{y}_k + \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} R_{k,g,t} \hat{x}_{g,t} \right) \\ +& \frac{\rho}{2} \sum_{k \in \mathcal{K}} \| z_k - D'_k \hat{y}_k + \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} R_{k,g,t} \hat{x}_{g,t} \|^2, \end{aligned} \tag{17}$$

where $\rho > 0$ is the penalty parameter and $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_K]$ are the lagrangian multipliers. We define $I_{iter}$ as the total number of iterations of the algorithm. In each iteration $i$, ADMM updates each variable block as follows (in line 5) and update multipliers (in line 6):

$$\hat{\mathbf{x}}_t^{i+1} = \underset{\hat{\mathbf{x}}_t \in \mathcal{X}_t}{\operatorname{argmin}} L(\hat{\mathbf{x}}_1^i, ..., \hat{\mathbf{x}}_T^i, \hat{\mathbf{y}}^i, \mathbf{z}^i, \boldsymbol{\lambda}^i), \ \forall t \in \mathcal{T}, \tag{18}$$

$$\hat{\mathbf{y}}^{i+1} = \underset{\hat{\mathbf{y}} \in \mathcal{Y}}{\operatorname{argmin}} L(\hat{\mathbf{x}}_1^i, ..., \hat{\mathbf{x}}_T^i, \hat{\mathbf{y}}^i, \mathbf{z}^i, \boldsymbol{\lambda}^i), \tag{19}$$

$$\mathbf{z}^{i+1} = \underset{\mathbf{z} \in \mathcal{Z}}{\operatorname{argmin}} L(\hat{\mathbf{x}}_1^i, ..., \hat{\mathbf{x}}_T^i, \hat{\mathbf{y}}^i, \mathbf{z}^i, \boldsymbol{\lambda}^i), \tag{20}$$

where $\mathcal{X}_t = \{\mathbf{x}_t | (14b), (14c), (14d)\}$, $\mathcal{Y} = \{\mathbf{y} | 0 \leq y_k \leq 1\}$ and $\mathcal{Z} = \{\mathbf{z} | z_k \leq 0\}$. When ADMM terminates, the continuous solution $\hat{x}_{g,t}$ is obtained in line 8. The rounding process is then carried out in lines 10-13 to convert the largest $\hat{x}_{g,t}$ in each time slot to 1 (selecting the most promising group $g$ for each $t$) and keep others 0.

The developed ADMM-HEU can provide sub-optimal benchmarks within an acceptable time span, since the subproblems in (18)-(20) can be solved in a parallel manner and with a smaller size than the original problem. However, ADMM-HEU requires $\mathcal{O}(1/\epsilon^2)$ iterations to achieve $\epsilon$-optimality, where $\epsilon$ is set as $\frac{\mu}{T(T+3)}$ [35]. At each iteration, we can solve the $T+2$ variable blocks by B&B with the time complexity of $\mathcal{O}(T \cdot 2^G + 2 \cdot 2^K)$. Thus, the total complexity is given by $\mathcal{O}(T^5 \cdot 2^G + T^4 \cdot 2^K)$, which might not sufficient for fast adaptation to network variations.

**Algorithm 1** ADMM-HEU

**input:** $D_k$, $D'_k$ and $R_{k,n,t}$.
Relax $\mathbf{P1}$ to a continuous problem $\mathbf{P1}'$.
Initialize $\hat{\mathbf{x}}_t^0$, $\hat{\mathbf{y}}^0$, $\mathbf{z}^0$, $\boldsymbol{\lambda}^0$ and $i = 0$.
**for** $i = 0, ..., I_{iter}$ **do**
  Update $\hat{\mathbf{x}}_t, \hat{\mathbf{y}}$ and $\mathbf{z}$ by Eq. (18), (19) and (20).
  $\lambda_k^{i+1} = \lambda_k^i + \rho \left( z_k^i - D'_k \hat{y}_k^i + \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} R_{k,g,t} \hat{x}_{g,t}^i \right)$.
**end for**
Obtain relaxed solution $\hat{x}_{g,t}$.
**for** $t \in \mathcal{T}$ **do**
  Find $g^\dagger = \underset{g \in \mathcal{G}}{\operatorname{argmax}} \{\hat{x}_{1,t}, ..., \hat{x}_{G,t}\}$.
  Set $x^*_{g^\dagger,t} = 1$ and $x^*_{g,t} = 0, \ \forall g \neq g^\dagger$.
**end for**
Calculate $y_k^*$ based on Eq. (13).
**output:** $x^*_{g,t}$ and $y_k^*$

### B. Conventional Online-Learning Solutions and Limitations

To enable an intelligent and online solution, we address the problem from an RL perspective. Firstly, we briefly introduce actor-critic and meta-critic learning approaches as a basis to present the proposed EMCL. AC is an RL algorithm that takes advantage of both value-based methods, e.g., Q-learning, and policy-based methods, e.g., REINFORCE, with fast convergent properties and the capability to deal with continuous action spaces [36]. The learning agent in AC contains two components, where the actor is responsible for making decisions while the critic is used for evaluating the decisions by the value functions. Specifically, at each learning step $t$[1], the actor takes action based on a stochastic policy, i.e., $a_t \sim \pi(a|s_t)$, where $\pi(a|s_t)$ is the probability of taking an action under state $s_t$, typically following the Gaussian distribution [37]. The critic is to generate a Q-value function $Q(s_t, a_t) = \mathbb{E}_\pi[\bar{r}_t|s_t, a_t]$, where $\bar{r}_t$ is the accumulated reward at step $t$, and $\mathbb{E}_\pi[\beta]$ is the expected value of $\beta$ over the policy $\pi$. The goal of the learning agent is to find a policy to maximize the expected accumulated reward (or Q-value).

A critical issue in conventional learning approaches, including AC, is that the performance of a learning model largely depends on the adopted training or observed data sets. To illustrate the dynamic environment and its impacts, we consider two types of environmental changes. The first is "foreseen variations". A typical example is a time-varying channel with certain time correlation and statistical characteristics. In this case, a general machine learning algorithm can capture the regular patterns effectively to resolve the mapping from the environment to the desired decision variables. The second is "unforeseen variations", which is much more challenging to address. These changes are usually unexpected and inclined to break the statistical distribution of the original environment. The practical LEO-5G systems are highly complex and dynamic, such as fast and dramatic variations in channel states, user demands, user arrival/departure, and network topologies. This typically causes the new inputs to no longer be relevant to the statistical properties of the historical data [38]. As a consequence, the scheduling decisions made from the previous learning model can become invalid and the model may need to be re-trained to adapt to the new environment. To illustrate this impact, we use Fig. 2, as an example, to depict a typical evolution of AC's loss value over time-varying demands. From 0 to 100 time slots, the demand is time-varying but follows historical statistical properties, e.g., fluctuating within a certain range or following a certain distribution, leading to a well-adapted AC with low and stable loss values. When a surged demand is generated at the 100-th time slot, the new input deviates from the statistics. The AC model becomes inapplicable to the new environment, evidenced by the rapidly deteriorating loss values. When the agent in AC consumes a considerable amount of time in new data collection and re-training, the performance can return to the previous level.

[1]In this paper, a learning step corresponds to a time slot.

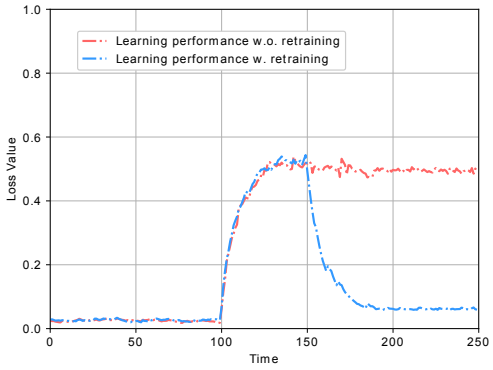



Fig. 2. Evolution of loss over time-varying demands.

To address this issue of "unforeseen change", meta-critic-based approaches become an emerging technique that takes advantage of a variety of previously observed tasks to infer the meta-knowledge, such that a new learning task can be quickly trained with few observations [15]. Meta-critic learning combines meta-learning with an AC framework to enhance the generalization ability. However, conventional meta-critic learning is not effective in dealing with the large discrete space in $\mathbf{P1}$. In addition, there is no uniform standard to parameterize the learning model and extract meta-knowledge in dynamic environments. Thus, we propose an EMCL algorithm to enable an efficient dynamic-adaptive solution.

## IV. The Proposed EMCL Algorithm

In this section, we elaborate the proposed EMCL algorithm, firstly starting from outlining the EMCL framework, then detailing the tailored design.

### A. EMCL Framework

*1) MDP Reformulation:* First, we reformulate the original problem $\mathbf{P1}$ as an MDP by defining action, state and reward.

- As the actor is to select a group from set $\mathcal{G}$ at each time slot $t$, the action is defined as an assigned link group,

$$a_t = g \in \mathcal{G}. \tag{21}$$

- The state consists of the channel coefficients $h_{k,n,t}$, modeled as FSMC with the transition probability defined

in (6), and the delivered data for user $k$ up to time slot $t$, where $b_{k,t} = b_{k,t-1} + R_{k,a_t,t}$.

$$s_t = \{h_{1,1,t}, ..., h_{K,N,t}, b_{1,t}, ..., b_{K,t}\}. \tag{22}$$

All possible states are included in the state space $\mathcal{S}$. The next state only depends on the current state and action but is irrelevant to the past, which means the state transition from $s_t$ to $s_{s+1}$ follows the Markov property [36].

- The reward is closely related to the objective of $\mathbf{P1}$. We define the reward as (23).

$$r_t = \sum_{k=0}^{K} \eta_k (\Delta_{k,t-1}^2 - \Delta_{k,t}^2), \tag{23}$$

where $\Delta_{k,t} = \begin{cases} \sum_{k=1}^{K} \mathbb{1}\left(b_{k,t} - D'_k\right) - K, & k = 0, \\ b_{k,t} - D_k, & k \neq 0. \end{cases}$

Then, the accumulated reward at step $t$ is given by $\bar{r}_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$, where $\gamma \in [0, 1]$ is a discounted factor.

Under the designed MDP, we verify the consistency between the goals of the RL algorithm and the original optimization problem such that the policy provided by the learning agent can minimize the objective in $\mathbf{P1}$.

**Lemma 2.** *When $\gamma = 1$, the objective of the learning agent is equivalent to that of the optimization problem* $\mathbf{P1}$.

*Proof.* See Appendix B □

*2) Meta Critic and Task-Specific Actor:* As shown in Fig. 3, we design a hierarchical structure in EMCL containing a meta critic[2] and multiple actors. Meta-learning uses data from previously observed multiple tasks, $\mathcal{J}^{(1)}, ..., \mathcal{J}^{(I)}$, to infer a "meta-knowledge" with good generalization ability and accelerate the training for a new task. In the proposed EMCL, the "meta-knowledge" is the meta critic which can evaluate the task with a Q-value, like the role of the critic in traditional AC, and possesses a strong generalization ability to guide any task-specific actor to provide a policy.

At time step $t$, $s_t^{(i)}$, $a_t^{(i)}$, and $r_t^{(i)}$ represent the state, action, and reward for task $i$, respectively. An episode $\mathcal{D}^{(i)} = \{s_1^{(i)}, a_1^{(i)}, r_1^{(i)} ..., s_T^{(i)}, a_T^{(i)}, r_T^{(i)}\}$ can be sampled from the first step to the terminal step $T$. We denote $\mathcal{D}_{[u,w]}^{(i)}$ as a segment of $\mathcal{D}^{(i)}$ from step $u$ to $w$, i.e., $\mathcal{D}_{[u,w]}^{(i)} = \{s_u^{(i)}, a_u^{(i)}, r_u^{(i)}, ..., s_w^{(i)}, a_w^{(i)}, r_w^{(i)}\}$. Since the explicit meta critic and actors are difficult to obtain, we adopt the function approximation method. The meta critic is parameterized as a neural network (NN) with the weights $\boldsymbol{\omega}$, i.e., $Q(s_t^{(i)}, a_t^{(i)}, \mathcal{D}_{[t-\bar{t},t-1]}^{(i)}; \boldsymbol{\omega})$. We note that, in addition to $s_t^{(i)}$ and $a_t^{(i)}$, the input includes the most recent $\bar{t}$ samples $\mathcal{D}_{[t-\bar{t},t-1]}^{(i)}$. Each task-specific actor is modeled as an NN $\pi(a|s_t^{(i)}; \boldsymbol{\theta}^{(i)})$ with the weights $\boldsymbol{\theta}^{(i)}$.

To optimize the weights, we minimize the loss functions by gradient descent. The loss function of the meta critic $L(\boldsymbol{\omega})$ is

[2]In this paper, "meta-critic learning" refers to an algorithm that combines AC and meta-learning while "meta critic" refers to the critic in the framework.

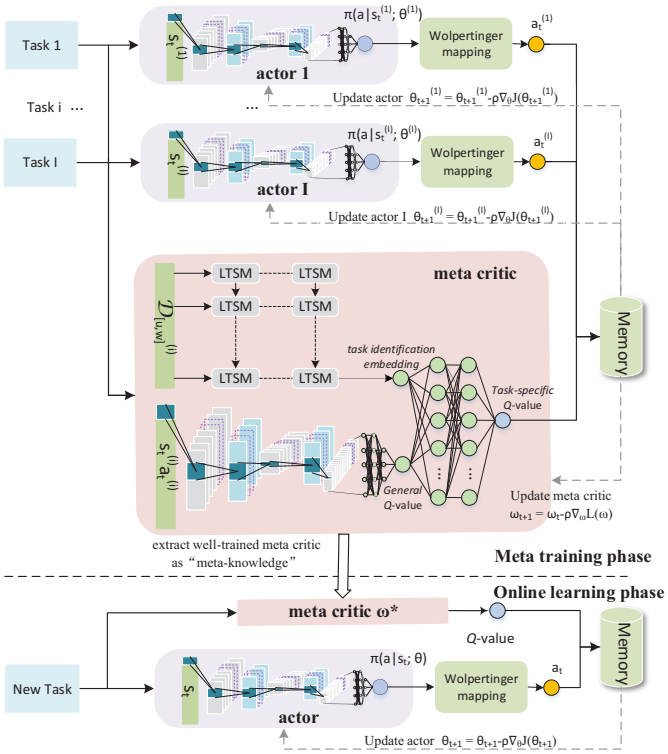



Fig. 3. The proposed EMCL framework.

defined as the average temporal difference (TD) error over all tasks:

$$L(\boldsymbol{\omega}) = \frac{1}{I} \sum_{i=1}^{I} \mathbb{E}_{\pi(\boldsymbol{\theta}^{(i)})} \Big[ \big(Q(s_{t+1}^{(i)}, a_{t+1}^{(i)}, \mathcal{D}_{[t-\bar{t}+1,t]}^{(i)}; \boldsymbol{\omega}) - r_t - \gamma Q(s_t^{(i)}, a_t^{(i)}, \mathcal{D}_{[t-\bar{t},t-1]}^{(i)}; \boldsymbol{\omega}) \big)^2 \Big], \tag{24}$$

where the TD error reflects the similarity between the estimated Q-value and actual Q-value. For the task-specific actor, the loss function $J(\boldsymbol{\theta}^{(i)})$ is the negative Q-value:

$$J(\boldsymbol{\theta}^{(i)}) = \mathbb{E}_{\pi(\boldsymbol{\theta}^{(i)})} \left[ -Q(s_t^{(i)}, a_t^{(i)}, \mathcal{D}_{[t-\bar{t},t-1]}^{(i)}; \boldsymbol{\omega}) \right], \tag{25}$$

such that minimizing $J(\boldsymbol{\theta}^{(i)})$ is equivalent to maximizing the expected accumulated reward. The update rules are given by:

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \rho \nabla_{\boldsymbol{\omega}} L(\boldsymbol{\omega}), \tag{26}$$

$$\boldsymbol{\theta}_{t+1}^{(i)} = \boldsymbol{\theta}_t^{(i)} - \rho \nabla_{\boldsymbol{\theta}^{(i)}} J(\boldsymbol{\theta}^{(i)}). \tag{27}$$

Based on the fundamental results of the policy gradient theorem [36], the gradients of $L(\boldsymbol{\omega})$ and $J(\boldsymbol{\theta}^{(i)})$ are:

$$\nabla_{\boldsymbol{\omega}} L(\boldsymbol{\omega}) = \frac{1}{I} \sum_{i=1}^{I} \Big[ 2L(\boldsymbol{\omega}) \nabla_{\boldsymbol{\omega}} (Q(s_{t+1}^{(i)}, a_{t+1}^{(i)}, \mathcal{D}_{[t-\bar{t}+1,t]}^{(i)}; \boldsymbol{\omega}) - Q(s_t^{(i)}, a_t^{(i)}, \mathcal{D}_{[t-\bar{t},t-1]}^{(i)}; \boldsymbol{\omega})) \Big], \tag{28}$$

$$\nabla_{\boldsymbol{\theta}^{(i)}} J(\boldsymbol{\theta}^{(i)}) = -Q(s_t^{(i)}, a_t^{(i)}, \mathcal{D}; \boldsymbol{\omega}) \nabla_{\boldsymbol{\theta}^{(i)}} \log \pi(a|s_t^{(i)}; \boldsymbol{\theta}^{(i)}). \tag{29}$$

*3) Algorithm Summary:* We summarize the proposed EMCL in Alg. 2, which includes two phases: the meta training

phase and the online learning phase. For the former, the meta critic is trained over different learning tasks. At each learning episode, we sample $I$ learning tasks. We obtain the approximated Q-value (in line 6) and stochastic policy (in line 7) by the approximation functions. The final actions are determined by the Wolpertinger approach in line 8, which will be elaborated in the following subsection. In line 9, the memory is used to store the experienced learning tuples $\{s_t^{(i)}, s_{t+1}^{(i)}, a_t^{(i)}, r_t^{(i)}\}$. At each step, we extract a batch of tuples from the memory as the training data for updating $\boldsymbol{\omega}$ and $\boldsymbol{\theta}^{(i)}$ by (26) and (27) in line 10 and 12, respectively. In the online learning phase, given a new task, the well-trained meta critic $\boldsymbol{\omega}^*$ can be directly used to estimate the Q-value and only the actor needs to be re-trained. We note that the adaptation ability of the meta-learning algorithm depends on the completeness of the tasks provided in the meta-training phase. In general, it is not practical to collect all the possible environments. As an alternative, the selected tasks in the meta-training phase should keep the diversity and representativeness to achieve higher sampling efficiency.

**Algorithm 2** EMCL

**Meta training phase:**
1: **input**: Multiple task samples; initial $\boldsymbol{\omega}_0$.
2: **for** each learning episode **do**
3: Sample $I$ tasks and initialize $\boldsymbol{\omega}_0, \boldsymbol{\theta}_0^{(1)}, ..., \boldsymbol{\theta}_0^{(I)}$.
4: **for** each learning step $t$ **do**
5: **for** each task $i$ **do**
6: Obtain Q-value by the meta critic in (32).
7: Obtain stochastic policy by the actor in (34).
8: Take actions $a_t^{(i)}$ by the Wolpertinger approach.
9: Store tuples $\{s_t^{(i)}, s_{t+1}^{(i)}, a_t^{(i)}, r_t^{(i)}\}$ in the memory.
10: Take a batch of data and update $\boldsymbol{\theta}^{(i)}$ by (27).
11: **end for**
12: Update $\boldsymbol{\omega}$ by (26).
13: **end for**
14: **end for**
15: **output**: The well-trained meta critic $\boldsymbol{\omega}^*$.
**Online learning phase:**
16: **input**: A new task; initial $\boldsymbol{\theta}_0$; well-trained meta critic $\boldsymbol{\omega}^*$.
17: **for** each learning episode **do**
18: **for** each learning step $t$ **do**
19: Obtain Q-value by the meta critic in (32).
20: Obtain stochastic policy by the actor in (34).
21: Take an action $a_t$ by the Wolpertinger approach.
22: Store tuples $\{s_t, s_{t+1}, a_t, r_t\}$ in the memory.
23: Take a batch of data and update $\boldsymbol{\theta}$ by (27).
24: **end for**
25: **end for**
26: **output**: The optimal actor $\boldsymbol{\theta}^*$.

### *B. Tailored Designs in EMCL*

*1) Parameterization with Hybrid Neural Networks:* There is no uniform standard for parameterization in conventional meta-critic learning. Considering dynamic environments, the distribution of the new input data and the previous observations may deviate. Towards fast adaptation to the dynamic environment, the critic should be able to identify different tasks, where the information for task identification can be refined from the experienced data, which usually forms time-related series [17]. The widely used DNN might have limitations in efficiency and in mining features from time-series data due to the massive number of weights and feed-forward structure. In the proposed EMCL, we design tailored neural networks to enable the meta critic and the actors to fit the complex nonlinear relationships and extract the meta-knowledge from historical data.

As shown in Fig. 3, for the meta critic, a hybrid neural network (HNN) combing convolutional neural network (CNN), long-short term memory (LSTM), and artificial neural network (ANN) is applied to learn the features from the current state-action pairs and historical trajectories [39]. Thereinto, CNN is computation-efficient via adopting the parameter sharing and pooling operations, and is effective to extract spatial features from the input data. These advantages enable CNN to reduce the parameters of the model and alleviate the problem of overfitting. LSTM, as a type of recurrent neural network, has advantages in extracting features from time-related sequential data. Thus, in the designed meta critic, the CNN is used to evaluate the decisions made by the actor from the current action-state pair $s_t^{(i)}, a_t^{(i)}$. The LSTM is adopted to identify the task based on the time-series data $\mathcal{D}_{[t-\bar{t},t-1]}^{(i)}$, such that the meta critic can accurately criticize any actor in changing environment and adapt to the dynamic networks. We denote $f_{cnn}(\mathbf{x};\mathbf{w})$, $f_{lstm}(\mathbf{x};\mathbf{w})$ and $f_{ann}(\mathbf{x};\mathbf{w})$ as the outputs of CNN, LSTM, and ANN, respectively, which are the functions of input $\mathbf{x}$ and weight $\mathbf{w}$. The features output from CNN and LSTM are:

$$\xi_1 = f_{cnn}(s_t^{(i)}, a_t^{(i)}; \boldsymbol{\omega}_{cnn}), \tag{30}$$

$$\xi_2 = f_{lstm}(\mathcal{D}_{[t-\bar{t},t-1]}^{(i)}; \boldsymbol{\omega}_{lstm}), \tag{31}$$

where $\xi_1$ and $\xi_2$ physically mean the general Q-value and the task identification embedding, respectively, which can be represented by scalars [17]. Then, we take the features as inputs and pass them through a fully-connected ANN to obtain the task-specific Q-value:

$$Q^{\pi}(s_t^{(i)}, a_t^{(i)}, \mathcal{D}_{[t-\bar{t},t-1]}^{(i)}; \boldsymbol{\omega}) = f_{ann}(\xi_1, \xi_2; \boldsymbol{\omega}_{ann}). \tag{32}$$

For the task-specific actors, we adopt CNN as the approximator which takes the current state as the input and outputs the mean $\mu$ and variance $\vartheta^2$ of the stochastic policy. We assume the stochastic policy follows Gaussian distribution $N(\mu, \vartheta^2)$, such that

$$[\mu, \vartheta^2] = f_{cnn}(s_t^{(i)}; \boldsymbol{\theta}^{(i)}), \tag{33}$$

$$\pi(a|s_t^{(i)}; \boldsymbol{\theta}^{(i)}) = N(\mu, \vartheta^2). \tag{34}$$

*2) Action Mapping with the Wolpertinger Policy:* The decision variables in $\mathbf{P1}$ are discrete such that we need to map the action from the stochastic policy to a discrete action space. However, the previous action mapping policies in meta-critic learning are not efficient since the action space is large for $\mathbf{P1}$. Thus, in EMCL, the Wolpertinger policy is adopted for faster convergence [40].

Following the stochastic policy $\pi$, the actor first produces an action $\hat{a}$ with continuous value, i.e.,

$$f_{\pi}: \mathcal{S} \rightarrow \hat{\mathcal{A}}, \quad f_{\pi}(s) = \hat{a}, \tag{35}$$

where $f_\pi$ is a mapping from the state space $\mathcal{S}$ to a continuous action space $\hat{\mathcal{A}}$ under the policy $\pi$. As the real action space $\mathcal{G}$ is discrete in $\mathbf{P1}$, the following two conventional approaches can be used for discretization [36]:

- Simple approach: $a_s^* = \underset{a\in\mathcal{G}}{\operatorname{argmin}} |a - \hat{a}|^2$.
- Greedy approach: $a_g^* = \underset{a\in\mathcal{G}}{\operatorname{argmax}} Q(s,a)$.

The simple approach is to select the closest integer value to $\hat{a}$. This approach may result in a high probability of deviating from the optimum, especially at the beginning of learning, and further lead to slow convergence [36]. The greedy approach optimizes Q-value at each step but the complexity is proportional to the exponentially increasing space $\mathcal{G}$ [36]. To achieve a trade-off between the complexity and learning performance, the Wolpertinger mapping approach is considered.

- Wolpertinger approach: $a_w^* = \underset{a\in\mathcal{M}^*}{\operatorname{argmax}} Q(s,a)$,

where $\mathcal{M}^*$ is a subset of $\mathcal{G}$ and contains $M$ nearest neighbors of $\hat{a}$. In the Wolpertinger approach, the final action is determined by selecting the highest-scoring action from $\mathcal{M}^*$. The Wolpertinger mapping becomes the greedy approach and simple approach when $M = |\mathcal{G}|$ and $M = 1$, respectively, and the solution of simple approach $a_s^*$ is included in $\mathcal{M}^*$.

**Lemma 3.** *We assume* $\mathcal{M}^* = \{a_1, ..., a_M\}$ *and* $\begin{cases} Q(s,a_m) \sim U(Q(s,a_s^*) - \kappa, Q(s,a_s^*) + \kappa), & m \neq m' \\ Q(s,a_m) = Q(s,a_s^*), & m = m', \end{cases}$ *where* $U(a,b)$ *refers to uniform distribution and* $\kappa$ *is a constant, then*

$$\mathbb{E}\left[Q(s,a_w^*)\right] = Q(s,a_s^*) + \kappa\left(1 - \frac{2(2^M-1)}{M\cdot 2^M}\right). \quad (36)$$

*Proof.* See Appendix C □

From Lemma 3, when $M > 1$, $\mathbb{E}\left[Q(s,a_w^*)\right] > Q(s,a_s^*)$, which means that the Wolpertinger approach finds the actions with higher Q-values than the simple approach at each learning step, and a larger $M$ leads to a higher expected Q-value. In addition, the complexity of the Wolpertinger approach is lower than the greedy approach as the size of the searching space decreases from $|\mathcal{G}|$ to $|\mathcal{M}^*|$. Thus, for the problems with huge discrete spaces, the Wolpertinger approach enables fast convergence to the maximum Q-value with a proper $M$.

### C. Complexity Analysis for EMCL

For the meta critic, an HNN, composed of CNN, LSTM, and ANN, is employed to estimate the Q-value. We assume CNN includes $V_1$ convolutional layers. We denote $o_{c,v}$, $o_{k,v}$, $o_{f,v}$ are the number of convolutional kernels, the spatial size of the kernel, and the spatial size of the output feature map in the $v$-th layer, respectively. The stripe of kernel is 1, and the input size is $o_{c,0} = K(N+1)+1$. The time complexity of CNN is $\mathcal{O}\left(\sum_{v=1}^{V_1} o_{c,v-1}\varrho_v\right)$, where $\varrho_v = o_{k,v}^2 o_{c,v} o_{f,v}^2$ [41]. For the LSTM, we consider $V_2$ layers, and denote $o_{l,v}$ and $o_{e,v}$ are the input size and number of memory cells for layer $v$, respectively, where $o_{l,0} = m(K(N+1)+1)$. The time complexity is given by $\mathcal{O}\left(\sum_{v=1}^{V_2} o_{e,v}(4o_{l,v-1} + \varsigma_v)\right)$, where $\varsigma_v = 4o_{e,v} + o_{l,v} + 3$ [42]. For the fully-connected ANN, the time complexity is $\mathcal{O}\left(2o_{d,1} + \sum_{v=2}^{V_3} o_{d,v-1}o_{d,v}\right)$, where $V_3$ is the number of layers of ANN, $o_{d,v}$ is the input size for layer $v$ [43]. For the actor, as the stochastic policy is approximated by a CNN, the time complexity is identical to that of CNN in the meta critic. Overall, the total time complexity of EMCL is calculated by $\mathcal{O}\left(TK(N+1)L_1 + L_2\right)$, where $L_1 = \varrho_1 + 4mo_{e,1}$ and $L_2 = \varrho_1 + o_{e,1}(4m+\varsigma_1) + 2o_{d,1} + \sum_{v=2}^{V_1} o_{c,v-1}\varrho_v + \sum_{v=2}^{V_2} o_{e,v}(4o_{l,v-1}+\varsigma_v) + \sum_{v=2}^{V_3} o_{d,v-1}o_{d,v}$. When the parameters of the learning model are determined, the complexity increases linearly with $\mathbf{P1}$'s input size, i.e., $K$ and $N$.

## V. Numerical Results

In the simulation, the adopted parameters for implementing EMCL are summarized in Table I. We compare the performance of the proposed EMCL algorithm with the following five benchmark algorithms:

- OPT: optimal solution (B&B).
- ADMM-HEU: suboptimal solution (Alg. 1).
- GRD: a greedy suboptimal algorithm proposed in [44].
- AC-DDPG: a classic AC algorithm with deep deterministic policy gradient proposed in [45].
- AC-MAML: AC with model-agnostic meta-learning proposed in [16].

The first three provide benchmarks from an optimization perspective, while the last two compare with EMCL from a learning perspective. For the AC benchmarks, the actor and critic are parameterized by two DNNs with the complexity $\mathcal{O}(TK(N+1)L_3 + L_4)$, where $L_3$ and $L_4$ are constants, thus keeping the same magnitude with the proposed EMCL [43].

We remark that although the formulated problem P1 is for resource allocation in one scheduling cycle, i.e., $T$ time slots, it can be extended to evaluate the average performance over the long term with multiple scheduling cycles. In simulations, if the original demand is not completely transmitted within one cycle, the demand can be updated by $D_k = D_k - R_k + \hat{D}_k, \forall k \in \mathcal{K}$, where $R_k = \sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} R_{k,g,t}x_{g,t}$ is the transmitted data in this scheduling cycle and $\hat{D}_k$ is the newly arrived demand of $k$. In the next cycle, P1 can be resolved with the updated demands. This process repeats until scheduling terminates.

### A. Capability in Dealing with Dynamic Environments

To verify the capability of the proposed EMCL in dealing with dynamic environments, Fig. 4-6 compare EMCL with AC-MAML and AC-DDPG in three dynamic scenarios. In Fig. 4, we consider the first scenario with users' irregular access and departure, which can be disruptive to the typical statistical properties. For instance, the adopted simulator generates user arrivals by following the Poisson distribution as the normal case, while it also periodically generates abnormal events (every 200 slots) with randomly large/small number of arrived users. We update the environment information every 200 time slots. From Fig. 4, both EMCL and AC-MAML are able to converge before each update, but EMCL saves 28.66% recovery time and reduces 45.42% objective value than AC-MAML, where we define a recovery time counting from

Table I: Parameter setting

| | |
|---|---|
| Total number of GDs in network | 500-1000 |
| Number of transmitters | 1 LEO, 1 BS and 2 TSTs |
| Time limitation $T$ | 10 time slots |
| Duration of time slot $\Phi$ | 0.1 s |
| $\varepsilon$ in $D_k' = \varepsilon D_k$ | 0.3 - 0.6 |
| Altitude of LEO | 780 km |
| Transmit power of LEO | 100 W |
| Transmit power of BS | 40 W |
| Transmit power of TST | 2 W |
| Bandwidth for C-band | 20 MHz |
| Bandwidth for Ka-band | 400 MHz |
| Carrier frequency of C-Band | 4 GHz |
| Carrier frequency of Ka-Band | 30 GHz |
| Noise power spectral density | -174 dBm/Hz |
| Weights values | $0 \leq \eta_0 \leq 10$<br>$\eta_1 + ... + \eta_K = 1$ |
| Parameterized meta critic | HNN |
| Parameterized actor | CNN |
| Distribution of stochastic policy | Gaussian |
| Learning rate | 0.001 |
| Batch size | 128 |
| Memory size | 10,000 |
| Discount factor | 0.9 |
| Size of search space in Wolpertinger policy | 10 |
| Environment update interval | 200 time slots |
| Software platform | Python 3.6 with TensorFlow 1.12.0 |

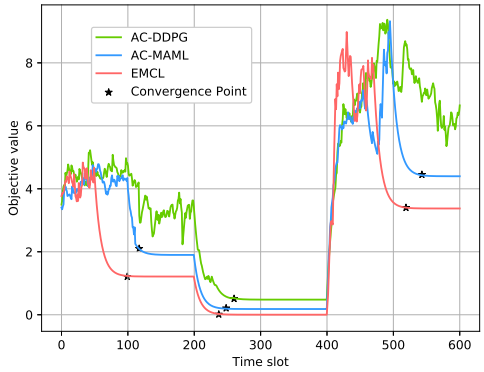



Fig. 4. Performance in adapting to dynamic scenario 1: user entry and leave.

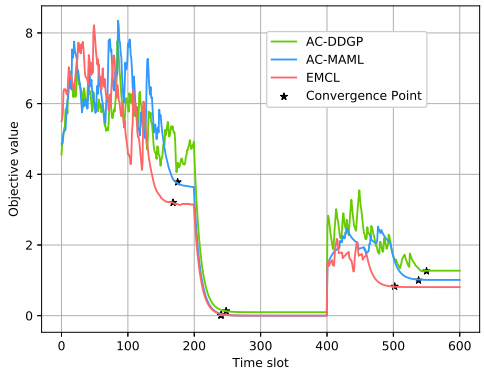



Fig. 5. Performance in adapting to dynamic scenario 2: bursty demands.

the moment of dramatic performance degradation until the performance recoveries to the normal level. For AC-DDPG, the convergence performance is inferior to the others, and fails to converge when updating occurs at the 200-th and 600-th time slot. We remark that the case of user departure is easier to be adapted. Fewer users in the system reduce the problem dimension, and thus simplify the learning task, leading to a halved recovery time and flat curves between the 200-th and the 400-th slots in three algorithms. In contrast, it is more difficult to deal with the case of user arrival, referring to the large fluctuation after the 400-th slot, mainly due to lacking relevant new-user data and the exponentially increasing dimension. We can observe that EMCL has strong capabilities in adapting to this difficult case and achieves more performance gains than the other two algorithms.

In Fig. 5, we evaluate the algorithms' capabilities in adapting to unforeseen dynamic demands. The simulator generates the volume of users' arrived demand by the uniform distribution as the normal case. Then, the distribution can be changed due to the abnormal bursty demands, e.g., switching from a low-speed voice call to a data-hungry HD video service, or vice versa. In Fig. 6, we consider the channel states can undergo non-ideal large fluctuations, e.g., sharply deteriorated channel conditions due to the large obstacles or the rain/cloud blocks appearing in the transmission path. Similarly to Fig. 4, we collect the updated environment information every 200 time slots. From Fig. 5 and Fig. 6, AC-DDPG has poor convergence performance, since AC-DDPG needs to re-train the learning model from scratch when the environment changes, leading to a slow adaptation, while EMCL and AC-MAML extract the meta-knowledge from multiple tasks to accelerate the convergence speed. EMCL re-fits the learning model in a timely manner than AC-MAML. This is because EMCL uses meta critic to guide the actor to adjust scheduling schemes more effectively in a dynamic environment, and the designed HNN and Wolpertinger mapping approach can improve the learning accuracy and efficiency in large discrete action spaces.

Fig. 7 further summarizes the average recovery time with respect to the numbers of GDs based on Fig. 4. In general, the more GDs in the system, the longer the recovery time required to adapt to the new environment. On average, EMCL saves 29.83% and 13.49% recovery time compared to AC-DDPG and AC-MAML, respectively, and the time-saving gain of EMCL becomes even larger when more GDs in the system. In addition, we compare the EMCL algorithm with and without the Wolpertinger policy to demonstrate the effectiveness of the adopted action mapping method. The recovery time of the latter is 10.11% increased than the former but less than AC-DDPG and AC-MAML. At the convergence, EMCL can decrease the average objective value by 30.36% compared to EMCL without the Wolpertinger policy.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3178171

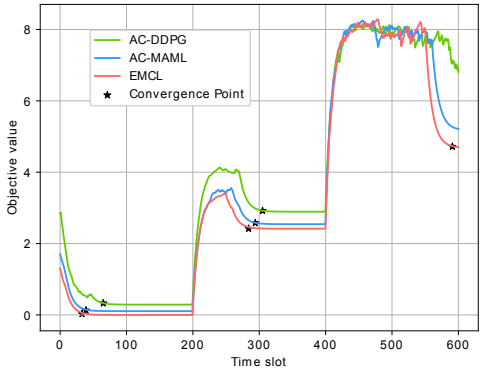



Fig. 6. Performance in adapting to dynamic scenario 3: unforeseen channel variations.

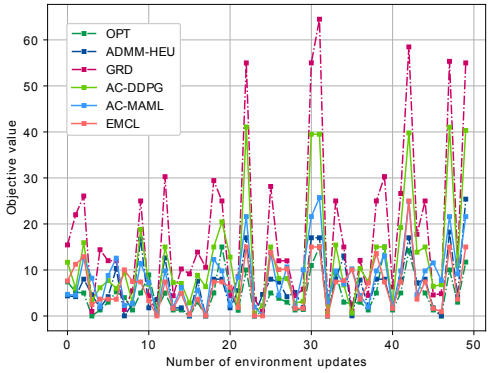



Fig. 8. Objective value vs. number of updates

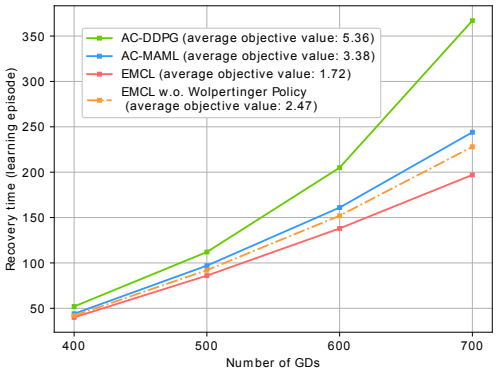



Fig. 7. Recovery time vs. number of users

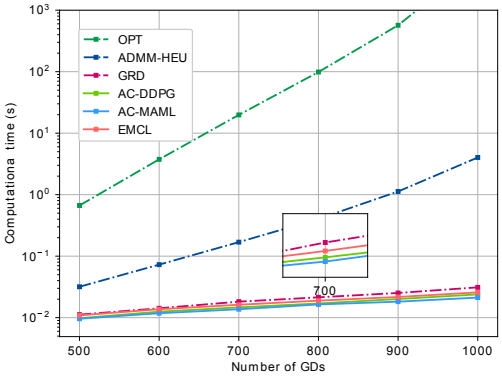



Fig. 9. Computational time vs. number of users

### *B. Trade-Offs between Computational Time and Optimality*

To demonstrate EMCL's trade-off performance between approaching the optimum (Fig. 8) and computational time (Fig. 9), we compare EMCL with five benchmarks. In Fig. 8, we observe 50 environmental information updates and record the average objective values within each update cycle. For AC-MAML and AC-DDPG, the average gaps to the optimum are 45.26% and 57.23%, respectively, while for EMCL, the average gap drops to 27.58%. The performance of EMCL is slightly better than ADMM-HEU, around 3.54%. For GRD, the average gap to the optimum is 74.15%, which is inferior to the AC-based algorithms.

Fig. 9 compares the computational time with respect to the number of GDs. OPT is the most time-consuming algorithm, as expected. Compared to OPT, ADMM-HEU saves 98.14% computational time by decomposing variables into multiple blocks and performing parallel computations. The computational time in ECML, two AC algorithms, and GRD keep at the millisecond level, but the proposed EMCL achieves smaller gaps to the optimum, hence concluding the better trade-off performance of EMCL than other benchmarks.

## VI. Conclusion

We have investigated a resource scheduling problem in dynamic LEO-terrestrial communication systems to address the mismatch issue in a practical over-loaded scenario. Due to the high computational time of the optimal algorithm and the proposed ADMM-HEU algorithm, we solve the problem from the perspective of DRL to obtain online solutions. To enable the learning model to fast adapt to dynamic environments, we develop an EMCL algorithm that is able to handle the environmental changes in wireless networks, such as bursty demands, users' entry/leave, and abrupt channel change. Numerical results show that, when encountering an environmental variation, EMCL consumes less recovery time to re-fit the learning model, compared to AC-DDPG and AC-MAML. Furthermore, EMCL achieves a good trade-off between solutions quality and computation efficiency compared to offline and AC-based benchmarks. An extension of the current work is to combine other techniques, e.g., continuous learning and behavior regularization, to further improve the sample efficiency and model adaptability.

## Appendix A
## Proof of Lemma 1

We relax all the binary variables of $\mathbf{P1}$ to continuous variables $\hat{\mathbf{x}} = [\hat{x}_{1,1}, ..., \hat{x}_{g,t}, ..., \hat{x}_{G,T}]^{\mathrm{T}}$ and $\hat{\mathbf{y}} =$

$[\hat{y}_1, ..., \hat{y}_k, ..., \hat{y}_K]^{\mathrm{T}}$, where $\hat{x}_{g,t}, \hat{y}_k \in [0,1]$. The relaxed objective function is written by:

$$f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \eta_0 \left(\mathbf{1}^{\mathrm{T}}\hat{\mathbf{y}} - K\right)^2 + \sum_{k\in\mathcal{K}} \eta_k \left(\mathbf{r}_k^{\mathrm{T}}\hat{\mathbf{x}} - D_k\right)^2, \quad (37)$$

where $\mathbf{1} = [1, ..., 1]^{\mathrm{T}}$ and $\mathbf{r}_k = [R_{k,1,1}, ..., R_{k,g,t}, ..., R_{k,G,T}]^{\mathrm{T}}$. We expand $\left(\mathbf{1}^{\mathrm{T}}\hat{\mathbf{y}} - K\right)^2$ and $\left(\mathbf{r}_k^{\mathrm{T}}\hat{\mathbf{x}} - D_k\right)^2$ as follows:

$$\left(\mathbf{1}^{\mathrm{T}}\hat{\mathbf{y}} - K\right)^2 = \hat{\mathbf{y}}^{\mathrm{T}}\mathbf{E}\hat{\mathbf{y}} - 2D_k\mathbf{1}_k^{\mathrm{T}}\hat{\mathbf{y}} + K, \quad (38)$$

$$\left(\mathbf{r}_k^{\mathrm{T}}\hat{\mathbf{x}} - D_k\right)^2 = \hat{\mathbf{x}}^{\mathrm{T}}\mathbf{R}\hat{\mathbf{x}} - 2D_k\mathbf{r}_k^{\mathrm{T}}\hat{\mathbf{x}} + D_k^2, \quad (39)$$

where $\mathbf{E}$ is an all-ones matrix and

$$\mathbf{R} = \begin{bmatrix} R_{k,1,1}^2 & R_{k,1,1}R_{k,1,2} & \cdots & R_{k,1,1}R_{k,G,T} \\ R_{k,1,2}R_{k,1,1} & R_{k,1,2}^2 & \cdots & R_{k,1,2}R_{k,G,T} \\ \vdots & \vdots & \ddots & \vdots \\ R_{k,G,T}R_{k,1,1} & R_{k,G,T}R_{k,1,2} & \cdots & R_{k,G,T}^2 \end{bmatrix}. \quad (40)$$

Referring to the theorem of quadratic programming, a quadratic function is convex when its corresponding real symmetric matrix is positive semi-definite [30]. According to the definition, $\mathbf{E}$ and $\mathbf{R}$ are positive semi-definite matrices since, given an arbitrary vector $\mathbf{v} = [v_1, ..., v_{G\times T}] \neq \mathbf{0}$, we can calculate $\mathbf{v}^T\mathbf{E}\mathbf{v} = \left(\mathbf{1}^{\mathrm{T}}\mathbf{v}\right)^2 \geq 0$ and $\mathbf{v}^T\mathbf{R}\mathbf{v} = \left(\mathbf{r}^{\mathrm{T}}\mathbf{v}\right)^2 \geq 0$ [30]. Therefore, $f(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is convex as it is the summation of $K+1$ convex functions. Besides, the constraints Eq. (14b)-(14d) are linear, hence the conclusion.

## Appendix B
## Proof of Lemma 2

The objective of the learning agent is to find a policy $\pi(a|s_t)$ that maximizes the expected accumulated reward $\sum_{t=0}^{T}\gamma^t r_t$. With $r_t$ in Eq. (23), we expand $\sum_{t=0}^{T}\gamma^t r_t$ as:

$$\begin{aligned}\sum_{t=0}^{T}\gamma^t r_t &= \sum_{k=0}^{K}\eta_k\left[\gamma\Delta_{k,0}^2 + \sum_{t=1}^{T}(\gamma^{t+1}-\gamma^t)\Delta_{k,t}^2 - \gamma^T\Delta_{k,T}^2\right] \\ &\overset{\gamma=1}{=} \sum_{k=0}^{K}\eta_k\left(\Delta_{k,0}^2 - \Delta_{k,T}^2\right) \\ &= \sum_{k=0}^{K}\eta_k\Delta_{k,0}^2 - \eta_0\left(\sum_{k=1}^{K}\mathbb{1}\left(b_{k,T} - D_k'\right) - K\right)^2 \\ &\quad - \sum_{k=1}^{K}\eta_k\left(b_{k,T} - D_k\right)^2, \end{aligned} \quad (41)$$

where $b_{k,T} = \sum_{t=1}^{T} R_{k,a_t,t}$. Thus, we can obtain the optimal policy $a_t^* \sim \pi^*(a|s_t)$ by solving the following problem:

$$\begin{aligned}\max_{\pi(a|s_t)} -\mathbb{E}_{\pi(a|s_t)}&\left[\eta_0\left(\sum_{k=1}^{K}\mathbb{1}\left(\sum_{t=1}^{T}R_{k,a_t,t} - D_k'\right) - K\right)^2\right. \\ &\left. - \sum_{k=1}^{K}\eta_k\left(\sum_{t=1}^{T}R_{k,a_t,t} - D_k\right)^2\right], \end{aligned} \quad (42)$$

which is equivalent to the objective Eq. (14a), thus the conclusion.

## Appendix C
## Proof of Lemma 3

Denote $Q(s,a_1), ..., Q(s,a_M)$ as random variables $X_1, ..., X_M$, where $X_{m'} = Q(s,a_s^*)$ and $X_m \sim U(Q(s,a_s^*)-\kappa, Q(s,a_s^*)+\kappa),\ \forall m \neq m'$. Thus, $Q(s,a_w^*)$ can be expressed as a random variable $\Psi = \max\{X_1, ..., X_M\}$. The cumulative distribution function of $\Psi$ is expressed as:

$$\begin{aligned}F_\Psi(\psi) =& \mathbb{P}[\Psi \leq \psi] = \mathbb{P}[\max\{X_1, ..., X_M\} \leq \psi] \\ =& \mathbb{P}[X_1 \leq \psi]\mathbb{P}[X_2 \leq \psi]...\mathbb{P}[X_M \leq \psi] \\ =& F_{X_1}(\psi)F_{X_2}(\psi)...F_{X_M}(\psi) \end{aligned} \quad (43)$$

For $m \neq m'$, based on the cumulative distribution function of uniform distribution, we can derive:

$$\begin{aligned}F_{X_m}(\psi) =& \frac{\psi - Q(s,a_s^*) + \kappa}{2\kappa}, \\ & \psi \in [Q(s,a_s^*) - \kappa, Q(s,a_s^*) + \kappa]. \end{aligned} \quad (44)$$

For $m = m'$, as $X_{m'} = Q(s,a_s^*)$, the cumulative distribution function is:

$$F_{X_{m'}}(\psi) = \mathbb{P}[X_{m'} \leq \psi] = \begin{cases} 1, & \psi \geq Q(s,a_s^*), \\ 0, & \psi < Q(s,a_s^*). \end{cases} \quad (45)$$

By substituting Eq. (44) and Eq. (45) into Eq. (43),

$$F_\Psi(\psi) = \begin{cases} \left(\frac{y - Q(s,a_s^*) + \kappa}{2\kappa}\right)^{M-1}, & \psi \in [Q(s,a_s^*), Q(s,a_s^*) + \kappa], \\ 0, & \psi \in [Q(s,a_s^*) - \kappa, Q(s,a_s^*)). \end{cases} \quad (46)$$

Then, the probability density function of $\Psi$ can be calculated by solving the first derivative:

$$\begin{aligned}&f_\Psi(\psi) = [F_\Psi(\psi)]' \\ =& \begin{cases} \frac{1}{2^{M-1}}\delta(\psi - Q(s,a_s^*)), & \psi = Q(s,a_s^*), \\ \frac{M-1}{2\kappa}\left(\frac{\psi - Q + \kappa}{2\kappa}\right)^{M-2}, & Q(s,a_s^*) < \psi \leq Q(s,a_s^*) + \kappa, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \quad (47)$$

where $\delta(\cdot)$ is Dirac function. The expectation of $\Psi$ is:

$$\begin{aligned}&\mathbb{E}[\Psi] = \mathbb{E}[Q(s,a_w^*)] = \int_{Q(s,a_s^*)}^{Q(s,a_s^*)+\kappa}\psi f_\Psi(\psi)d\psi \\ =& \frac{1}{2^{M-1}}\int_{Q(s,a_s^*)^-}^{Q(s,a_s^*)^+}\psi\delta(y - Q(s,a_s^*))dy \\ &+ \int_{Q(s,a_s^*)^+}^{Q(s,a_s^*)+\kappa}\psi\frac{M-1}{2\kappa}\left(\frac{\psi - Q(s,a_s^*) + \kappa}{2\kappa}\right)^{M-2}d\psi \\ =& \frac{Q(s,a_s^*)}{2^{M-1}} + Q(s,a_s^*) + \kappa - \frac{2\kappa}{M} - \frac{Q(s,a_s^*)}{2^{M-1}} + \frac{\kappa}{M\cdot 2^{M-1}} \\ =& Q(s,a_s^*) + \kappa\left(1 - \frac{2(2^M-1)}{M\cdot 2^M}\right). \end{aligned} \quad (48)$$

Thus the conclusion.

## References

[1] Y. Li, E. Pateromichelakis, N. Vucic, J. Luo, W. Xu, and G. Caire, "Radio resource management considerations for 5G millimeter wave backhaul and access networks," in *IEEE Communication Magazine*, vol. 55, no. 6, pp. 8692, Jun. 2017.

[2] O. Kodheli et al., "Satellite communications in the new space era: a Survey and future challenges," in *IEEE Communications Surveys and Tutorials*, vol. 23, no. 1, pp. 70-109, 2021.

[3] L. You, K. Li, J. Wang, X. Gao, X. Xia, and B. Ottersten, "Massive MIMO transmission for LEO satellite communications," in *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp.1851-1865, Jun. 2020.

[4] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense LEO: integrating terrestrial-satellite networks into 5G and beyond for offloading," in *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 47-62, Jan. 2019.

[5] Y. Li, N. Deng, and W. Zhou, "A hierarchical approach to resource allocation in extensible multi-layer LEO-MSS," in *IEEE Access*, vol. 8, pp.18522-18537, Jan. 2020.

[6] S. Wang, Y. Li, Q. Wang, M. Su, and W. Zhou, "Dynamic downlink resource allocation based on imperfect estimation in LEO-HAP cognitive system," in Proc. *IEEE International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct. 2019.

[7] J. H. Lee, J. Park, M. Bennis, and Y. C. Ko, "Integrating LEO satellite and UAV relaying via reinforcement learning for non-terrestrial networks," in Proc. *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2020.

[8] S. He, T. Wang, and S. Wang, "Load-aware satellite handover strategy based on multi-agent reinforcement learning," in Proc. *IEEE Global Communications Conference (GLOBCOM)*, Dec. 2020.

[9] B. Deng, C. Jiang, H. Yao, S. Guo and S. Zhao, "The next generation heterogeneous satellite communication networks: integration of resource management and deep reinforcement learning," in *IEEE Wireless Communications*, vol. 27, no. 2, pp. 105-111, Apr. 2020.

[10] L. Lei, Y. Yuan, T. X. Vu, S. Chatzinotas, M. Minardi, and J. F. Mendoza Montoya, "Dynamic-adaptive AI solutions for network slicing management in satellite-integrated B5G systems," in *IEEE Network Magazine*, 2021.

[11] Y. Shen, Y. Shi, J. Zhang and K. B. Letaief, "LORM: learning to optimize for resource management in wireless networks with few training samples," in *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 665-679, Jan. 2020.

[12] O. Simeone, S. Park and J. Kang, "From learning to meta-learning: reduced training overhead and complexity for communication systems," in *6G Wireless Summit (6G SUMMIT)*, pp. 1-5, 2020.

[13] H. Sun, W. Pu, M. Zhu, X. Fu, T. H. Chang and M. Hong, "Learning to continuously optimize wireless resource in episodically dynamic environment," in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4945-4949, 2021.

[14] K. Javed and M. White, " Meta-learning representations for continual learning," in Proc. *Neural Information Processing Systems (NIPS)*, pp. 1820-1830, Dec. 2019.

[15] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in Proc. *International Conference on Machine Learning (ICML)*, pp. 1126-1135, Jul. 2017.

[16] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in Proc. *International Conference on Machine Learning (ICML),* pp. 53315340, 2019.

[17] F. Sung, L. Zhang, T. Xiang, T. Hospedales, and Y. Yang, "Learning to learn: meta-critic networks for sample efficient learning," in *arXiv preprint arXiv:1706.09529*, Jun. 2017.

[18] H. Wu, Z. Zhang, C. Jiao, C. Li and T. Q. S. Quek, "Learn to sense: a meta-learning-based sensing and fusion framework for wireless sensor networks," in *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8215-8227, Oct. 2019.

[19] S. Park, O. Simeone and J. Kang, "Meta-learning to communicate: fast end-to-end training for fading channels," in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5075-5079, 2020.

[20] 3GPP TR 23.737, "Study on architecture aspects for using satellite access in 5G (release 17)".

[21] M. Cheng, J. B. Wang, J. Cheng, J. Y. Wang and M. Lin, "Joint scheduling and precoding for mmwave and sub-6ghz dual-mode networks," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13098-13111, Nov. 2020.

[22] X. Fang, W. Feng, T. Wei, Y. Chen, N. Ge, and C. X. Wang, "5G embraces satellites for 6G ubiquitous IoT: Basic models for integrated satellite terrestrial networks,"in *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14399-14417, Mar. 2021.

[23] X. Zhu, C. Jiang, L. Kuang, N. Ge, and J. Lu, "Energy efficient resource allocation in cloud based integrated terrestrial-satellite networks." in Proc. *IEEE International Conference on Communications (ICC)*, pp. 1-6, May 2018.

[24] W. Wang, T. Chen, R. Ding, G. Seco-Granados, L. You, and X. Gao, "Location-based timing advance estimation for 5g integrated leo satellite communications," in *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6002-6017, Jun. 2021.

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

[26] "Doppler compensation, uplink timing advance and random access in NTN," 3GPP TSG RAN WG1 Meeting, no. 97, R1-1906087, May 2019.

[27] K. Guo et al., "Performance analysis of hybrid satellite-terrestrial cooperative networks with relay selection," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9053-9067, Aug. 2020.

[28] H. Wang and N. Moayeri, "Finite-State Markov Channel-A Useful Model for Radio Communication Channels," in *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163-171, Feb. 1995.

[29] L. Lei, D. Yuan, C. K. Ho and S. Sun, "Optimal Cell Clustering and Activation for Energy Saving in Load-Coupled Wireless Networks," in *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6150-6163, Nov. 2015.

[30] J. Nocedal and S. Wright "Numerical optimization," Springer Science & Business Media, 2006.

[31] "Digital Video Broadcasting (DVB); Implementation guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2: S2 Extensions (DVB-S2X)." DVB Document A171-2, Apr. 2020.

[32] 3GPP TR 36.776, "Study on LTE-based 5G terrestrial broadcast (release 16)".

[33] C. H. Papadimitriou and K. Steiglitz, "Combinatorial optimization: algorithms and complexity," Mineola, NY, USA: Dover, 1998.

[34] S. Boyd, N. Parikh, and E. Chu, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Hanover, MA, USA: Now Publishers Inc., 2011.

[35] T. Lin, S. Ma, and S. Zhang, "Iteration complexity analysis of multi-block ADMM for a family of convex minimization without strong convexity," in *Journal of Scientific Computing,* col. 69, no. 1, pp. 52-81, Oct. 2016.

[36] R. S. Sutton and A.G. Barto, "Reinforcement Learning: An Introduction", Cambridge, Massachusetts, USA: MIT Press, 2018.

[37] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: an actor-critic reinforcement learning approach," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680-692, Nov. 2017.

[38] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: a review," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346-2363, Dec. 2019.

[39] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning", London, MIT press, 2016.

[40] J. Ye, and H. Gharavi, "Deep reinforcement learning-assisted energy harvesting wireless networks," in *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 990-1002, Jun. 2021.

[41] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in Proc. *IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5353-5360, 2015.

[42] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in Proc. *International Speech Communication Association (ISCA)*, 2014.

[43] Y. Yuan, L. Lei, T. X. Vu, S. Chatzinotas, S. Sun, and B. Ottersten, "Energy minimization in UAV-aided networks: actor-critic learning for constrained scheduling optimization," in *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5028-5042, May 2021.

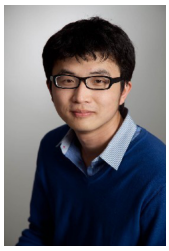
[44] Z. Jiang, S. Chen, S. Zhou, and Z. Niu, "Joint user scheduling and beam selection optimization for beam-based massive MIMO downlinks," in *IEEE Transactions on Wireless Communications,* vol. 17, no. 4, pp. 2190-2204, Jan. 2018.

[45] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in Proc. *Neural Information Processing Systems (NIPS)*, Jun. 2014.

**Yaxiong Yuan** (S'18) received the M.S. degree from the Laboratory of Wireless Communication Systems and Networks (WCSN), Beijing University of Posts and Telecommunications, Beijing, China. He is currently pursuing the Ph.D degree with Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg. His research interests include optimization theory, machine learning, wireless resource management, and wireless communication networks.

**Lei Lei** (S'12-M'17) is currently associate professor at Xi'an Jiaotong University, School of Information and Communications Engineering. He received the B.Eng. and M.Eng. degrees from Northwestern Polytechnic University, Xi'an, China, in 2008 and 2011, respectively. He obtained his Ph.D. degree in 2016 at the Department of Science and Technology, Linköping University, Sweden. He was with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg as a research associate and research scientist from 2016 to 2021. He was a research assistant at Institute for Infocomm Research (I2R), A*STAR, Singapore, in 2013. His current research interests include resource allocation and optimization in terrestrial-satellite networks, energy-efficient communications, and deep learning in wireless communications. He received the IEEE Sweden Vehicular Technology-Communications-Information Theory (VT-COM-IT) joint chapter best student journal paper award in 2014. He was a co-recipient of the IEEE SigTelCom 2019 Best Paper Award.

**Thang X. Vu** (S'11–M'15) received the B.S. and the M.Sc., both in Electronics and Telecommunications Engineering, from the VNU University of Engineering and Technology, Vietnam, in 2007 and 2009,respectively, and the Ph.D. in Electrical Engineering from the University Paris-Sud, France, in 2014. In 2010, he received the Allocation de Recherche fellowship to study Ph.D. in France. From September 2010 to May 2014, he was with the Laboratory of Signals and Systems (LSS), a joint laboratory of CNRS, Centrale Supelec and University Paris-Sud XI, France. From July 2014 to January 2016, he was a postdoctoral researcher with the Information Systems Technology and Design (ISTD) pillar, Singapore University of Technology and Design (SUTD), Singapore. Currently, he is a research scientist at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. His research interests are in the field of wireless communications, with particular interests of wireless edge caching, cloud radio access networks, machine learning for communications and cross-layer resources optimization. He was a recipient of the SigTelCom 2019 best paper award.



**Zheng Chang (S'10-M'13-SM'17)** received the B.Eng. degree from Jilin University, Changchun, China, in 2007, the M.Sc. (Tech.) degree from Helsinki University of Technology (Now Aalto University), Espoo, Finland, in 2009, and the Ph.D. degree from the University of Jyväskylä, Jyväskylä, Finland, in 2013.

Since 2008, he has been holding various research positions at Helsinki University of Technology, the University of Jyväskylä, and Magister Solutions Ltd., Finland. He was a Visiting Researcher at Tsinghua University, China, from June to August in 2013, and the University of Houston, TX, USA, from April 2015 to May 2015. He has published over 100 papers in journals and conferences. His research interests include the IoT, cloud/edge computing, security and privacy, vehicular networks, and green communications. He serves as a TPC Member for many IEEE major conferences, such as INFOCOM, ICC, and GLOBECOM. He has been awarded by Ulla Tuominen Foundation, Nokia Foundation, and Riitta and Jorma J. Takanen Foundation for his research excellence. He has been awarded as the 2018 IEEE Communications Society Best Young Researcher for Europe, Middle East, and Africa Region. He received the best paper awards from IEEE TCGCC and APCC in 2017. He was selected as an Exemplary Reviewer of IEEE WIRELESS COMMUNICATIONS LETTERS in 2018. He has participated in organizing workshop and special session in GLOBECOM'19, WCNC18–22, SPAWC'19, and ISWCS'18. He serves as an Editor for IEEE WIRELESS COMMUNICATIONS LETTERS, Wireless Networks (Springer), and International Journal of Distributed Sensor Networks. He serves as a Guest Editor for IEEE Network, IEEE WIRELESS COMMUNICATIONS, IEEE Communications Magazine, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, Physical Communication, EURASIP Journal on Wireless Communications and Networking, and Wireless Communications and Mobile Computing.

**Symeon Chatzinotas** (S'06–M'09–SM'13) is currently Full Professor/Chief Scientist I and Head of the SIGCOM Research Group with the SnT, University of Luxembourg. He coordinates research activities in communications and networking, acting as a PI in more than 20 projects and is the main representative for 3GPP, ETSI, DVB. In the past, he was a Visiting Professor with the University of Parma, Parma, Italy, lecturing on 5G Wireless Networks. He was involved in numerous Research and Development Projects for NCSR Demokritos, CERTH Hellas and CCSR, University of Surrey, Guildford, U.K. He has coauthored more than 500 technical papers in refereed international journals, conferences and scientific books. He was the corecipient of the 2014 IEEE Distinguished Contributions to Satellite Communications Award and Best Paper Awards at EURASIP JWCN, CROWNCOM, ICSSC, WCNC. He is currently on the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY, and the International Journal of Satellite Communications and Networking.

**Sumei Sun** (F'16) is a Principal Scientist and Head of the Communications and Networks Dept at the Institute for Infocomm Research ($I^2R$), Singapore. She is also holding a joint appointment with the Singapore Institute of Technology, and an adjunct appointment with the National University of Singapore, both as a full professor. Her current research interests are in next-generation wireless communications, cognitive communications and networks, and industrial internet of things. She is Editor-in-Chief of IEEE Open Journal of Vehicular Technology, member of the IEEE Transactions on Wireless Communications Steering Committee, and a Distinguished Speaker of the IEEE Vehicular Technology Society 2018-2021. She is also Director of IEEE Communications Society Asia Pacific Board and a member at large with the IEEE Communications Society.