Cost-Effective Hybrid Computation Offloading in Satellite–Terrestrial Integrated Networks

Xinyuan Zhang, *Graduate Student Member, IEEE*, Jiang Liu, Zehui Xiong, *Senior Member, IEEE*, Yudong Huang, *Graduate Student Member, IEEE*, Ran Zhang, *Member, IEEE*, Shiwen Mao, *Fellow, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—The Internet of Things (IoT) ecosystem is undergoing a significant evolution through its integration with satellite networks, empowering remote and computation-intensive IoT tasks to leverage computing services via satellite links. Current research in this field predominantly focuses on minimizing latency and energy consumption in computation offloading, yet overlooks the substantial costs incurred by satellite resource utilization. To address this oversight, we introduce a cost-effective hybrid computation offloading (CE-HCO) paradigm in satellite–terrestrial integrated networks (STINs) in this article. First, we propose the 5G-based system framework facilitates gNB and user plane function functionalities on satellites and fosters collaboration between public cloud providers and satellite operators. The framework is in line with the latest 3GPP activities and business models in satellite computing. Then, we formulate the CE-HCO problem, aiming to minimize total computation offloading costs while satisfying diverse user latency requirements and adhering to satellite energy constraints. To tackle this NP-hard problem, we develop an algorithm employing the penalty method and successive convex approximation to simplify the complex mixed-integer nonlinear programming into tractable convex iterations. Simulation results show that our approach outperforms existing baselines in balancing performance and cost, and offer guidance on pricing policies for satellite computing services to promote future commercial growth.**

***Index Terms*—Computation offloading, mobile edge computing, satellite–terrestrial integrated network (STIN), successive convex approximation (SCA).**

Manuscript received 10 November 2023; revised 24 February 2024 and 18 June 2024; accepted 3 July 2024. Date of publication 8 July 2024; date of current version 7 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U23B2025, and in part by the State Scholarship Fund of China Scholarship Council under Grant 202106470020. *(Corresponding author: Jiang Liu.)*

Xinyuan Zhang and Yudong Huang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: zhangxinyuan0181@bupt.edu.cn; hyduni@bupt.edu.cn).

Jiang Liu and Ran Zhang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Future Network Research Center, Purple Mountain Laboratories, Nanjing 211111, China (e-mail: liujiang@bupt.edu.cn; zhangran@bupt.edu.cn).

Zehui Xiong is with the Information Systems Technology and Design Pillar Department, Singapore University of Technology and Design, Singapore 487372 (e-mail: zehui_xiong@sutd.edu.sg).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea (e-mail: zhan2@uh.edu).

Digital Object Identifier 10.1109/JIOT.2024.3424782

I. Introduction

The Internet of Things (IoT) has dramatically transformed our daily lives and revolutionized industries by interconnecting massive devices [1]. As IoT technology has rapidly evolved in recent years, new applications have arisen, such as real-time cargo ship tracking [2], offshore energy exploration [3], and smart mining [4], illuminating a future of intelligent and autonomous IoT systems. Yet, many of these pivotal applications are generated in remote places without terrestrial network infrastructure. Satellite networks bridge this gap, providing seamless connections between IoT devices and cloud data centers for computation, which is called satellite-enabled cloud computing (SCC). A notable example is Google Cloud's collaboration with the Starlink constellation in 2021 to deliver globally accessible cloud services [5]. Moreover, recent advancements in low-Earth orbit (LEO) satellite onboard capabilities, as demonstrated in Table I, have made satellite edge computing (SEC) possible. This approach allows satellites to function as edge nodes, directly processing data and thus eliminating the long propagation delay and high-backhaul burden due to cloud transmission. Amazon Web service (AWS)'s successful deployment of an Earth image processing software on a satellite stands as a testament to this [6], having reduced the amount of imaging data sent back to the Earth by 42% [7]. The ongoing 3GPP Release 19 is also actively engaged in research to standardize onboard edge computing capabilities [8], [9].

In the computing paradigms of SCC and SEC, computation offloading in satellite–terrestrial integrated networks (STINs) is one of the most significant problems. Computation offloading intricately involves choosing an offloading destination (SCC or SEC), selecting the appropriate cloud center or satellite, and determining the right amount of the task to offload. To illustrate, Cheng et al. [10] delved into offloading tasks from remote base stations to cloud centers through satellite backhaul transmission. Zhang et al. [11] focused on offloading scheduling within SEC, while [1], [12], [13] offloaded tasks to both SCC and SEC. Typically, these studies aimed to minimize latency or energy consumption, simultaneously optimizing the allocation of resources like transmission power, bandwidth, and computation capacity.

While previous studies provided valuable insights on computation offloading in STINs, solely concentrating on latency or energy consumption minimization did not provide

TABLE I
DEVELOPMENT OF TYPICAL AEROSPACE CPU CHIPS [23]

	CPU chip	Year	Main frequency	Core number	Computation capability	Architecture
1	TSC695F	1998	25.0 MHz	1	20 DMIPS	SPARC
2	RAD750	2001	200.0 MHz	1	400 DMIPS	PPC
3	DAHLIA	2019	1.6 GHz	4	4000 DMIPS	ARM
4	HPSC	2019	800.0 MHz	8	7360 DMIPS	ARM
5	BM3883	2021	1.0 GHz	8	16 GIPS	SPARC
6	Yulong810A	2021	1.0 GHz	8	12 TOPS	ARM

a comprehensive view, especially considering the expensive nature of satellite resources. For instance, focusing only on minimizing latency will lead to prioritizing tasks processed on satellites to avoid the extended propagation delays to cloud centers. However, it is important to note that operating a satellite edge server is estimated to be at least three times higher than a terrestrial cloud center server [14]. As satellite operators typically have a cost budget, offloading decisions should lean more on a judicious balance between cost and latency than just latency. Additionally, energy consumption alone does not fully reflect the cost. Offloading decisions influence satellite's power supply, battery cell lifespan [11], and heat dissipation [15]. These factors collectively affect long-term satellite maintenance and operational expenses, and should thus be integrated into cost calculations. In general, with the satellite computing market rapidly growing, there is a critical demand for developing cost-effective computation offloading approaches in STINs to promote broader adoption of satellite computing.

In this article, we propose a cost-effective hybrid computation offloading (CE-HCO) paradigm in STINs. In particular, CE-HCO incorporates both SCC and SEC as a hybrid offloading approach and aims to minimize the total computation offloading cost while ensuring the heterogeneous latency requirements of various tasks and maintaining satellite energy consumption limits. The challenges and intricacies of scheduling CE-HCO include the following.

1) Despite extensive research on computation offloading in terrestrial edge-cloud collaborative networks, STINs present unique challenges. They inherently exhibit dynamic connectivity and channel statuses for both satellite–terrestrial and intersatellite links (ISLs). How to schedule computation offloading to counter these dynamics and ensure service continuity?
2) How to determine the most cost-effective offloading approach?
3) Scheduling CE-HCO is NP-hard. It requires a joint optimization of offloading choices (i.e., offloading the task to SEC, SCC, or both and identifying which cloud center or satellite to employ), routing schedules (i.e., designing the routing paths to the cloud center and among satellites), and resource allocation (i.e., balancing the use of satellite communication resource, satellite computation resource, and terrestrial computation resource). How to derive the solution of CE-HCO in low-computation complexity?

The major contributions of this article are summarized as follows.

1) We propose a 5G-based CE-HCO system framework, which encompasses the 5G-based protocol framework, network architecture, and workflows. The framework enables the functionalities of gNodeB (gNB) and user plane function (UPF) onboard, aligning with the latest 3GPP 5G nonterrestrial network (NTN) design principles, tailored to address the distinctive satellite challenges. It also accommodates the collaboration of public cloud providers and satellite operators and takes into account their varying pricing policies, positioning it in line with the cutting-edge business models in the satellite computing market.
2) We formulate a CE-HCO problem, which coordinates offloading decisions across multiple tasks both between and within SCC and SEC. The problem optimizes the offloading cost for both public cloud providers and satellite operators while guaranteeing user latency requirements within the resource and energy constraints of satellites. To solve this NP-hard problem, we utilize the penalty method and successive convex approximation (SCA) approach to transform the nonconvex mixed-integer programming problem into a successive convex one. We then propose an algorithm to derive the near-optimal solution of the CE-HCO problem. The computation complexity and convergence of the algorithm are analyzed.
3) Through simulations, we demonstrate the proposed algorithm's effectiveness in balancing offloading performance with cost, compared to baseline algorithms. Our findings also provide insight into various factors that influence computation offloading decisions, including pricing strategies, the device's distance to cloud centers, and task-specific details.

The remainder of this article is organized as follows. Related work is elaborated in Section II. The CE-HCO system framework is described in Section III. The modeling and formulation of CE-HCO problem are presented in Section IV. The problem solution is in Section V. Simulation results are discussed in Section VI and conclusions are provided in Section VII.

II. RELATED WORK

A. Satellite Computing—State-of-the-Art

Satellite networks have long been recognized as a highly promising communication paradigm, offering seamless global coverage and resilience against disasters. Hence, they have served as transparent pipes, relaying signaling and data traffic between terrestrial nodes and thus substantially reducing the need for fiber deployments in rural areas. The architecture of 3GPP NTN-based RAN with transparent satellite is depicted in [16]. In the last three years, major public cloud providers, such as Microsoft Azure, AWS, and Google, have collaborated with several satellite operators to deliver global cloud services [17]. By 2032, the global satellite computing market is estimated to reach $472.6 million. The academic community has also delved into SCC. For instance, De Sanctis et al. [18] provided an overview of satellite communications in the Internet of Remote Things (IoRT), discussing crucial topics like heterogeneous network interoperability, QoS management, group-based communications, and IPv6 support. Additionally, Chien et al. [19] presented a broader scenario where space networks comprising satellites at different altitudes could serve land/marine/aviation/space-based users as cloud computing access.

In recent years, with the rapid development of aerospace and information technology, the capabilities of space-grade CPUs, FPGAs, and other chips have seen significant improvements, as shown in Table I. With these advancements, SEC has garnered significant interest from both industry and academia. Companies like Loft Orbital are pioneering the development of onboard edge computing processors for military satellites [20]. Meanwhile, AWS successfully ran a machine-learning software suite on a satellite to analyze Earth images [6], [7]. 3GPP Release 19 is examining the feasibility of supporting edge computing capabilities onboard to further reduce data transmission latency and minimize the consumption of backhaul resources. Several new architectures for SEC have been proposed. For example, Xie et al. [21] outlined an SDN/NFV-based SEC architecture and delved into critical functional components. Cao et al. [22] designed hardware and software deployments on SEC platforms. Compared to the bent-pipe SCC, SEC offers many compelling benefits, such as less response delay, lighter backhaul strain, and enhanced data security, since it eliminates the need to forward all tasks to a cloud center.

B. Computation Offloading in STINs

Research on computation offloading in STIN typically falls into four categories: 1) relaying data to remote terrestrial cloud centers via satellite backhaul; 2) offloading tasks to multiple satellites through multiple ground-to-satellite links; 3) satellite peer offloading via ISLs; and 4) offloading to both satellites and the ground. The work in the first category assumes that satellites can forward radio signals but lack data processing capabilities. Cheng et al. [10] offloaded tasks generated from the ground and the air through satellite backhaul, with a focus on minimizing the satellite's energy consumption and transmission delay. In [13], remote terrestrial multiaccess edge computing (MEC) base stations connected to cloud computing services via satellite backhaul, enabling adaptable task division and cooperative computing strategies. Although this approach expands cloud computing service to all regions of the world, it suffers from long propagation delays and overburdens the uplinks/downlinks between satellites and the ground due to their limited bandwidth. The second category explores on-board data processing capabilities but only offloads tasks through multiple ground-to-satellite links. Song et al. [24] offloaded tasks from energy-constrained IoT devices to multiple satellites through several uplinks established by a terrestrial-satellite terminal (TST). Cao et al. [12] investigated the allocation of satellite communication and computation resources to minimize energy consumption. While this approach alleviates the propagation delay, it still introduces congestion in uplinks/downlinks between satellites and the ground. Moreover, the uplinks/downlinks are susceptible to weather conditions, which can make offloading prone to failure. The third category encompasses studies assuming all tasks offloaded to satellites through both intersatellite data forwarding and on-board processing. Zhang et al. [11] proposed a distributed computation offloading scheme among neighbor satellites, achieving load balancing while minimizing satellite energy consumption. The main challenge here is the inherent limitation of on-board energy and resources, leading to potential inefficiencies in task processing due to resource overutilization. For the last category, Chen et al. [1] tackled the multitier partial computation offloading problem to reduce end-to-end latency and energy consumption. Chai et al. [25] centered on a multitask offloading issue, considering the associative relationships between tasks. However, they lack focus on the overall offloading costs.

Our proposed method distinguishes itself by focusing on cost-effective hybrid offloading strategies, offering a practical and scalable solution amidst the rapidly evolving satellite computing market.

C. Cost-Effective Computation Offloading in Terrestrial Networks

Numerous studies have addressed computation offloading from economic perspectives in terrestrial MEC and cloud computing networks. Du et al. [26] jointly optimized offloading decisions, device clustering, and resource allocation in a nonorthogonal multiple access vehicle edge computing network, characterizing communication and computation resource expenses as costs. Wang et al. [27] considered the overall welfare of the MEC system alongside the payment decisions of individual users. Jiao et al. [28] delved into the tradeoffs between cloud/fog computing service providers and miners, designing an auction mechanism to maximize social welfare. In [29], an online multiround auction mechanism was developed for resource trading between edge clouds and mobile devices.

While these studies provide key insights into profit-maximization and cost-minimization in terrestrial networks, they cannot be directly applied to STINs due to unique challenges. Specifically, the high-speed movement of LEO satellites with respect to Earth leads to dynamic satellite-ground and intersatellite connections and network topologies, which are fundamentally different from static terrestrial networks. Therefore, a reconsideration of the computing offloading strategy in STINs is essential to prevent performance degradation and ensure continuous service. To address this challenge, we account for dynamic

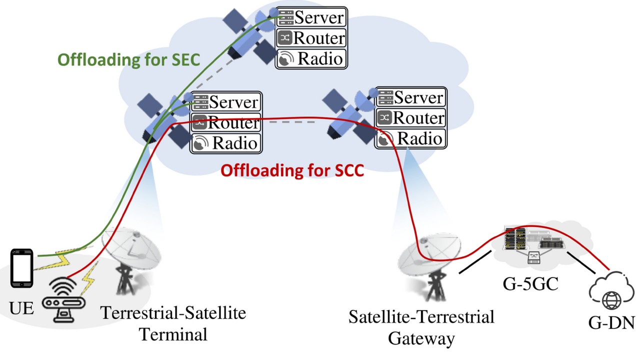
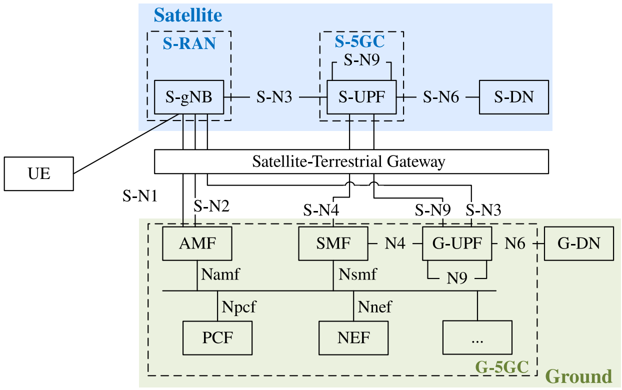


Fig. 1. 5G-based CE-HCO network architecture.



UE: User Equipment
S-RAN: Satellite Radio Access Network
S-5GC: Satellite 5G Core
S-DN: Satellite Data Network
S-UPF: Satellite User Plane Function
G-5GC: Ground 5G Core
G-UPF: Ground User Plane Function
G-DN: Ground Data Network
AMF: Access and Mobility Management Function
SMF: Session Management Function
PCF: Policy Control Function
NEF: Network Exposure Function

Fig. 2. 5G-based CE-HCO protocol framework.

satellite-ground channel status and visual relationships, introducing a snapshot model to discretize the long-term, ever-changing network status into manageable time slots. The channel model and visual relationship of satellite-ground links are considered for each slot.

III. 5G-Based CE-HCO Framework

CE-HCO is designed to provide pay-as-you-go computing services for tasks from remote regions without terrestrial network infrastructure. This is achieved through the collaboration of satellite operators and public cloud providers. In this section, we propose a 5G-based CE-HCO system framework to establish a foundation for the practical and feasible execution of efficient CE-HCO scheduling and implementation. The framework integrates a 5G-based protocol framework, network architecture, and workflows.

A. System Overview

Illustrated in Fig. 1, user equipments (UEs) include IoT devices and mobile devices in remote regions without terrestrial network infrastructure. UEs equipped with global navigation satellite system (GNSS) can know their location at any time. Considering the limited transmission capacity of IoT devices, a dedicated TST is deployed near IoT devices. The task bits of UEs within TST's small cell coverage will be first sent to the TST and then uploaded to satellites. The TST's wireless communication resource allocation among multiple IoT devices can refer to [24], which is not the focus of our study. The TST equipped with multiple independent antenna apertures can choose to establish connections with satellites in favorable weather conditions. This multiplicity in connections enhances uplink/downlink diversity, significantly boosting link availability under various weather conditions. Due to challenging rural deployment environments, the TST acts solely as an access point and not as a powerful MEC server because of constrained space and power supply [24]. Additionally, deploying MEC servers is not economically viable given the sparse distribution of IoT devices [30].

Moreover, the CE-HCO network consists of a satellite segment built upon emerging mega LEO constellations managed by operators like SpaceX and a ground segment supported by public cloud providers like Amazon AWS. In the satellite segment, each satellite performs radio base station, router, and server functionalities. Specifically, the radio base station capabilities equip the satellite to function as an S-gNB in Fig. 2, localizing radio processing in line with 3GPP Release 19. Meanwhile, the router functionalities facilitate the onboard S-UPF for efficiently networking via ISLs. Additionally, the server functionalities empower satellites to serve as edge computing servers within S-DN, adhering to the guidelines set forth in Release 18 and TR 23.700. In the ground segment, the dynamic pricing information of satellite operators and cloud providers is stored in G-5GC. CE-HCO scheduling is executed in G-5GC, and the signaling messages related to CE-HCO routing is also initiated by the G-5GC.

CE-HCO supports three computation offloading methods, each with unique network performance and associated costs. First, in SCC, raw input data and output results traverse long-distance ISLs between devices and cloud centers. This multihop forwarding takes long propagation delays and significant bandwidth costs but incurs minimal terrestrial computing expenses. Second, in SEC, tasks are distributed to several nearby satellites for parallel computation, reducing the propagation delay and communication expenses, but at a higher cost on satellite computing. Third, the joint SCC and SEC approach can facilitate parallel processing across satellite and ground computing, optimizing costs and maintaining acceptable delays. A significant problem here is determining the offloading distribution between SEC and SCC, which will be explored in the subsequent section.

B. System Workflow

Based on the system framework, the detailed 5G-based CE-HCO workflow is as follows.

1) *Initial Registration:* When the UE registers to 5G for the first time, it sends a registration request to G-5GC. AMF authenticates the UE and notifies SMF with QoS/billing profiles. Besides, our design assumes the adoption of geospatial IP addressing for the satellite network, as described in [31].

2) *Session Establishment:* The UE sends the session request to G-5GC. Based on the satellite network's status and the UE's QoS profile, either the PCF or SMF determines the offloading policy. This policy is derived by solving the CE-HCO optimization problem based on the real-time pricing strategies of operators, as presented in the next section. Subsequently, based on the decision, SMF selects a S-UPF as the session anchor and establishes traffic steering rules based on IP addresses.
3) *Offloading:* Traffic designated for SCC offloading travels from the UE to the access S-gNB via the S-N1 interface, then to the access satellite's S-UPF via S-N3, is forwarded across multiple S-UPFs via S-N9, and finally reaches the G-DN through N6. If offloaded to SEC, the task is routed to the appropriate SEC servers through a multihop path among S-UPFs via S-N9 and then through the S-N6 interface between the S-UPF and S-DN.

IV. Problem Formulation

In this section, we first present the network model. Next, we introduce the coverage and communication models for satellite-ground links, which differ from those of terrestrial networks. Then, we formulate the latency, energy consumption, and cost models in CE-HCO. Finally, we define the cost-effective CE-HCO problem.

A. Network Model

The constellation period, which refers to the repeated cycles for satellites as observed from the Earth surface, is split into multiple time slots, each indexed by $t \in \mathcal{T} = \{1, \ldots, T\}$. In each slot, the network topology is considered static, referred to as a snapshot [12], [32], [33]. For a given time slot t, the snapshot topology is represented by $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t\}$, where $\mathcal{V}_t$ denotes the vertex set and $\mathcal{E}_t$ denotes the edge set. In subsequent discussions, we will describe $\mathcal{G}_t$ of a single snapshot as an example.

The vertex set is defined as $\mathcal{V}_t = \mathcal{I}_t \cup \mathcal{S}_t \cup \mathcal{GS}_t$. Here, $\mathcal{I}_t$ represents the set of IoT devices, with each device indexed by $i \in \{1, \ldots, I_t\}$. $\mathcal{S}_t$ denotes the set of satellites, with each satellite indexed by $s \in \{1, \ldots, S_t\}$. It is assumed that all satellites are under the ownership of a single satellite operator. The set $\mathcal{GS}_t$ designates the ground stations, with each indexed by $g \in \{1, \ldots, GS_t\}$, and all are owned by a single cloud provider. The edge set, $\mathcal{E}_t$, comprises all links between the vertices in $\mathcal{V}_t$ that can establish connections within t.

Given the high velocity of LEO satellites, it is practical to treat the IoT devices as stationary within the duration of a single constellation snapshot. Assuming that each i generates at most one task in t, we interchangeably use i to represent devices and tasks in the subsequent discussions. Task i is characterized by the tuple $\{b_i, h_i, d_i\}$, where b_i is the task's size in bits, h_i denotes the required number of CPU cycles for processing one bit of task, and d_i denotes the task's deadline. We suppose the deadlines are smaller than the duration of a time slot. The ratios of task i processed by SEC and by SCC are given by α_i^S and α_i^G, respectively.

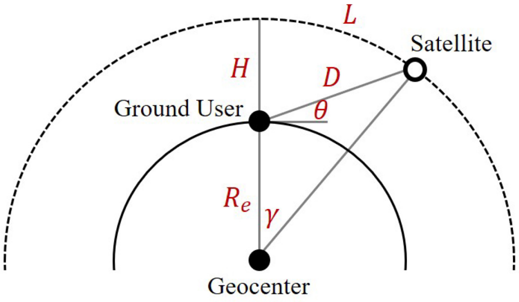


Fig. 3. Visual relationship between satellites and ground users.

At the beginning of each slot t, the PCF or SMF determines the offloading results for the tasks arrived in last slot. In subsequent discussions, we will describe the offloading scheduling a single slot as an example, and will therefore omit the subscript t for simplicity.

B. Coverage and Communication Models for Satellite-Ground Links

Different from terrestrial MEC systems, satellites move rapidly, causing time-varying changes in the visual relationship between satellites and ground users. Below, we describe the coverage model for satellite-ground links.

According to [34], the visual relationship between satellites and ground users is illustrated in Fig. 3. In this figure, H is the satellite orbit height, R_e is the Earth radius, D is the distance between the ground user and the satellite, and θ is the elevation angle, which can be determined by

$$\theta = \arccos\left(\frac{R_e + H}{D} \cdot \sin\gamma\right)$$

where γ is the geocentric angle and can be expressed as

$$\gamma = \arccos\left(\frac{R_e}{R_e + H} \cdot \cos\theta\right) - \theta.$$

Then, we can obtain the longest arc length that the satellite is within the ground user's line of sigh

$$L = 2 \cdot (R_e + H) \cdot \gamma.$$

Hence, the longest coverage time between the ground user and the satellite can be calculated by

$$T_v = \frac{L}{v}$$

where v is set based on ground terminal and constellation settings. As shown above, the coverage time between any ground user and satellite pair can be precalculated. This means that the visual relationship during any given time slot t can be predicted. If the coverage duration for a ground user and satellite pair does not fully fall within a time slot t, we assume they cannot establish communication links during that slot.

Each IoT device accesses satellites through the TST. According to [35], the satellite's Ka-band spectrum allocated to TST is B. Operating on the Ka-band, the TST's antenna

TABLE II
MAJOR NOTATIONS

Notation	Description
$\mathcal{T}, t$	The set of time slots and the index of time slots
H, R_e, θ, γ	Constellation parameters: the orbit height, the earth radius, the elevation angle, and the geocentric angle
D, T_v	The distance and longest coverage time between the ground user and satellite pair
$\mathcal{I}, i, I$	The set of devices, the index of devices, and the number of devices
$\mathcal{S}, s, S$	The set of satellites, the index of satellites, and the number of satellites
$\mathcal{GS}, g, GS$	The set of ground stations, the index of ground stations, and the number of ground stations
α_i^S, α_i^G	The ratio of task i processed by SEC, and by SCC
$\beta_{i,s}, \gamma_{i,s}$	The ratio of task i workload offloaded to satellite s and the 0-1 variable indicating whether task i is offloaded to satellite s in SEC
b_i, h_i, d_i	Task i's size, required number of CPU cycles for processing one bit, and the deadline
ρ	The ratio of output data size to input data size
$x_{i,s}^S, x_{i,s}^G, y_{i,s}, z_{i,s}$	The 0-1 variable indicating whether satellite s is the intermediate node on task i's SEC offloading path and SCC offloading path, whether satellite s is task i's access satellite, and the satellite nearest to the ground station on task i's offloding path
s_0, s_g	Task i's access satellite and the satellite nearest to the ground station on task i's offloading path
$T_{i,s}^{S,com}$	The computation latency of satellite s for processing task i in SEC
$T_{i,s_0}^{S,trans}, T_{s_0,i}^{S,trans}, T_{s_0,s}^{S,trans}, T_{s,s_0}^{S,trans}$	The transmission delay of uplink, downlink, and ISLs in SEC
$T_{i,s_0}^{S,prop}, T_{s_0,i}^{S,prop}, T_{s_0,s}^{S,prop}, T_{s,s_0}^{S,prop}$	The propagation delay of uplink, downlink, and ISLs in SEC
$T_{i,s_0}^{G,trans}, T_{s_0,i}^{G,trans}, T_{s_g,g}^{G,trans}, T_{g,s_g}^{G,trans}, T_{s_0,s_g}^{G,trans}, T_{s_0,s_g}^{G,trans}$	The transmission delay of uplink (downlink) TST-satellite links, uplink (downlink) satellite-ground station links, and ISL (s_0, s_g) and (s_g, s_0) in SCC
$T_{i,s_0}^{G,prop}, T_{s_0,i}^{G,prop}, T_{s_g,g}^{G,prop}, T_{g,s_g}^{G,prop}, T_{s_0,s_g}^{G,prop}, T_{s_g,s_0}^{G,prop}$	The propagation delay of uplink (downlink) TST-satellite links, uplink (downlink) satellite-ground station links, and ISL (s_0, s_g) and (s_g, s_0) in SCC
$T_i^{S,Off}, T_i^S, T_i^S$	The total delay of satellite peer computation offloading in SEC, the total delay in SEC and SCC
$R_{TST,s_0}, R_{s_0,TST}, R_{ISL}, R_{s_g,g}, R_{g,s_g}$	The transmission rate of uplink (TST, s_0), downlink (s_0, TST), ISL, downlink (s_g, g),and uplink (g, s_g)
$hop_{s_0,s}, hop_{s_0,s_g}$	The hop count of the offloading path from satellites s_0 to s and from satellites s_0 to s_g
$D_{TST,s_0}, D_{s_0,TST}, D_{s_0,s}, D_{s_g,g}, D_{s_0,s_g}$	The distance between the TST and satellite s_0, satellites s_0 and s, satellite s_g and GS g, satellites s_0 and s_g
$E_{i,s}^{S,com}$	The computation energy consumption of satellite s processing i
$E_{i,s}^{S,trans}, E_{i,s}^{G,trans}$	The transmission energy consumption of satellite s for task i in SEC and SCC
P_{ISL}^T, P_{down}^T	The transmit power of satellites for ISLs and satellite downlinks
$p_C, p_{up}, p_{down}, p_{ISL}$	The price of satellite computing resource, satellite-terrestrial uplink, downlink, ISL communication resources
C_i^{com}, C_i^{trans}	The cost of using satellite computing and communiation resources for task i
C	The total cost of offloading all tasks
$r^{s,max}, E^{s,max}$	The maximum computing resources/energy that satellite s can provide

array has good directivity, making the sidelobe leakage tolerable. Therefore, TST's intersatellite interference can be ignored. The transmission rate from TST to satellite s can be given by

$$R_{\mathrm{TST},s} = B\log_2\left(1 + \frac{p_{\mathrm{TST}} G_{\mathrm{TST},s} h_{\mathrm{TST},s}}{\sigma_s^2}\right)$$

where p_{TST} is the transmit power of TST and $G_{\mathrm{TST},s}$ is the antenna gain of TST toward satellite s. The channel gain between the TST and satellite s is denoted as $h_{\mathrm{TST},s}$. Specifically, we have $h_{\mathrm{TST},s} = D_{\mathrm{TST},s}^{-\varepsilon}$, where $D_{\mathrm{TST},s}$ is the distance between TST and satellite s and ε represents the path loss exponent. The noise variance is denoted as σ_s^2. Note that, due to the relatively slow speed of the satellite compared with the large distance between the TST and the satellite, large-scale fading predominantly impacts the signal strength, while the influence of small-scale fading can be neglected [35]. The major notations are listed in Table II.

C. Latency and Energy Consumption in SEC

1) Latency: For task i, the transmission and propagation latency of the uplink to the access satellite s_0 are[1]

$$T_{i,s_0}^{S,\mathrm{trans}} = \frac{\alpha_i^S b_i}{R_{\mathrm{TST},s_0}},\ T_{i,s_0}^{\mathrm{prop}} = \frac{D_{\mathrm{TST},s_0}}{c}$$

where c is the light speed. Here, since the IoT devices are sparsely distributed around the TST and IoT tasks are periodically collected, the TST is assumed to reserve sufficient bandwidth to these devices. Further optimization of device-to-TST transmission latency can refer to [24], which is not our focus. Similarly, given that ρ is the ratio of output data size to input data size and $R_{s_0,\mathrm{TST}}$ is the transmission rate of downlink, the downlink transmission propagation latency are

$$T_{s_0,i}^{S,\mathrm{trans}} = \frac{\rho\, \alpha_i^S b_i}{R_{s_0,\mathrm{TST}}},\ T_{s_0,i}^{\mathrm{prop}} = T_{i,s_0}^{\mathrm{prop}}.$$

[1] We assume that a TST can only access at most one satellite at a time. However, our approach can be adjusted to accommodate scenarios where a TST can access multiple satellites for diversity.

After the task is successfully uploaded through the uplink, it is distributed among multiple peer satellites (i.e., $s \in \mathcal{S}$) for parallel processing. The computation delay of satellite s for processing task i is

$$T_{i,s}^{S,\text{com}} = \frac{\beta_{i,s}\alpha_i^S b_i}{r_i^s}$$

where $\beta_{i,s} \in [0, 1]$ is the proportion of workload $\alpha_i^S b_i$ offloaded to satellite s and r_i^s is the computation capacity of satellite s allocated to task i. We suppose that a satellite's computation capacity is divided into multiple computing resource blocks of equal size[2] [38]. A task can be allocated with at most one computing resource block at a time. All computation capacity of one resource block is taken by task i.

The transmission delay of task i's input data from the access satellite s_0 to the computing satellite s is

$$T_{s_0,s}^{S,\text{trans}} = \frac{\beta_{i,s}\alpha_i^S b_i \text{hop}_{s_0,s}}{R_{\text{ISL}}}$$

where $\text{hop}_{s_0,s}$ is the hop count of the offloading path from the access satellite s_0 to satellite s and R_{ISL} is the transmission rate of ISLs. Similarly, the output data's transmission delay is $\rho T_{s_0,s}^{S,\text{trans}}$. Although the Doppler shift caused by adjacent satellites moving with different relative velocities can result in data loss, it is easy to predict and calculate once the constellation parameters, as well as the velocity and position of any satellite at any time, are known in advance. Consequently, the Doppler shift between any pair of satellites during any period can be predicted and compensated using efficient methods, such as optical phase-lock loops and others [11]. Therefore, we assume that the satellite receivers are equipped with Doppler shift frequency compensators and exclude the effect of Doppler shift on ISL transmission. The propagation delay of the input data's offloading path from the access satellite s_0 to satellite s is

$$T_{s_0,s}^{S,\text{prop}} = \frac{D_{s_0,s}}{c}$$

where $D_{s_0,s}$ is the length of the offloading path. Similarly, the output data's propagation delay is equal to $T_{s_0,s}^{S,\text{prop}}$.

Hence, we denote $T_i^{S,\text{Off}}$ as the total latency of satellite peer computation offloading in SEC

$$T_i^{S,\text{Off}} = \max_{s\in\mathcal{S}}\left\{T_{i,s}^{S,\text{com}} + (1+\rho)T_{s_0,s}^{S,\text{trans}} + 2\gamma_{i,s}T_{s_0,s}^{S,\text{prop}}\right\} \quad (1)$$

where $\gamma_{i,s}$ is a 0-1 variable to indicate whether task i is offloaded to satellite s.

The total latency in SEC is

$$T_i^S = T_{i,s_0}^{S,\text{trans}} + T_{i,s_0}^{\text{prop}} + T_i^{S,\text{Off}} + T_{s_0,i}^{S,\text{trans}} + T_{s_0,i}^{\text{prop}}. \quad (2)$$

2) Energy Consumption: Now, we present the energy consumption of each satellite in SEC. First, if satellite s is one of task i's computation node, the computation energy consumption of satellite s for processing task i is

$$E_{i,s}^{S,\text{com}} = \kappa\beta_{i,s}\alpha_i^S b_i\left(r_i^s\right)^2 \quad (3)$$

where κ is a coefficient depending on the chip architecture.

If satellite s is the intermediate node on the offloading path of task i (but not the access satellite), the transmission energy consumption of satellite s for task i is

$$E_{i,s}^{S,\text{trans}} = \frac{(1+\rho)\alpha_i^S b_i P_{\text{ISL}}^T}{R_{\text{ISL}}}$$

where P_{ISL}^T is the transmission power of satellite transmitters for ISLs. For simplicity, we assume that each intermediate node forwards the same amount of workload $\alpha_i^S b_i$.

If s_0 is the access satellite of device i, the transmission energy consumption is

$$E_{i,s_0}^{S,\text{trans}} = \frac{\alpha_i^S b_i P_{\text{ISL}}^T}{R_{\text{ISL}}} + \frac{\rho\alpha_i^S b_i P_{\text{down}}^T}{R_{s_0,\text{TST}}}$$

where P_{down}^T is the transmission power of satellite transmitters for downlinks.

Hence, the total transmission energy consumption of satellite s for task i in SEC is

$$E_{i,s}^{S,\text{trans}} = x_{i,s}^S\frac{(1+\rho)\alpha_i^S b_i P_{\text{ISL}}^T}{R_{\text{ISL}}} + y_{i,s}\left(\frac{\alpha_i^S b_i P_{\text{ISL}}^T}{R_{\text{ISL}}} + \frac{\rho\alpha_i^S b_i P_{\text{down}}^T}{R_{s,\text{TST}}}\right) \quad (4)$$

where $x_{i,s}^S$ is a 0-1 variable indicating whether satellite s is the intermediate node on SEC's offloading path and $y_{i,s}$ is a 0-1 variable indicating whether satellite s is device i's access satellite.

D. Latency and Energy Consumption in SCC

In SCC, i will first access satellite s_0 through the TST, then the input data $\alpha_i^G b_i$ will be forwarded through ISLs to satellite s_g, which is the nearest satellite to the ground station. The output data is also sent back to i through the reverse path.

1) Latency: Similar to SEC, the uplink/downlink transmission delay and uplink/downlink propagation delay of ground-satellite links are as follows:

$$T_{i,s_0}^{G,\text{trans}} = \frac{\alpha_i^G b_i}{R_{\text{TST},s_0}}, T_{s_0,i}^{G,\text{trans}} = \frac{\rho\alpha_i^G b_i}{R_{s_0,\text{TST}}}$$
$$T_{i,s_0}^{G,\text{prop}} = T_{s_0,i}^{G,\text{prop}} = \frac{D_{\text{TST},s_0}}{c}.$$

Moreover, the uplink/downlink transmission latency and uplink/downlink propagation latency of satellite-ground station links are

$$T_{s_g,g}^{G,\text{trans}} = \frac{\alpha_i^G b_i}{R_{s_g,g}}, T_{g,s_g}^{G,\text{trans}} = \frac{\rho\alpha_i^G b_i}{R_{g,s_g}}, T_{s_g,g}^{G,\text{prop}} = T_{g,s_g}^{G,\text{prop}} = \frac{D_{s_g,g}}{c}.$$

The transmission latency of ISLs between satellites s_0 and s_g is

$$T_{s_0,s_g}^{G,\text{trans}} = \frac{\alpha_i^G b_i \text{hop}_{s_0,s_g}}{R_{\text{ISL}}}, \quad T_{s_g,s_0}^{G,\text{trans}} = \rho T_{s_0,s_g}^{G,\text{trans}}$$

[2] For example, a satellite is equipped with an edge server. Several virtual machines are deployed in the edge server. Each virtual machine installing the edge computing software is a resource block [36], [37].

where hop_{s_0,s_g} is the hop count of the offloading path from satellites s_0 to s_g. The propagation delay of ISLs between satellites s_0 and s_g is

$$T^{G,\text{prop}}_{s_0,s_g} = T^{G,\text{prop}}_{s_g,s_0} = \frac{D_{s_0,s_g}}{c}.$$

Considering the powerful computing capability of the cloud center, we ignore its computing latency [1]. Hence, the total latency in SCC is

$$\begin{aligned} T^G_i = T^{G,\text{trans}}_{i,s_0} + T^{G,\text{trans}}_{s_0,i} + 2T^{G,\text{prop}}_{i,s_0} + T^{G,\text{trans}}_{s_g,g} \\ + T^{G,\text{trans}}_{g,s_g} + 2T^{G,\text{prop}}_{s_g,g} + (1+\rho)T^{G,\text{trans}}_{s_0,s_g} + 2T^{G,\text{prop}}_{s_0,s_g}. \end{aligned} \tag{5}$$

2) *Energy Consumption:* Similar to (4), each satellite's energy consumption during the transmission to the cloud center is

$$\begin{aligned} E^{G,\text{trans}}_{i,s} &= x^G_{i,s}\frac{(1+\rho)\alpha^G_i b_i P^T_{\text{ISL}}}{R_{\text{ISL}}} \\ &\quad + + y_{i,s}\left(\frac{\alpha^G_i b_i P^T_{\text{ISL}}}{R_{\text{ISL}}} \frac{\rho\alpha^G_i b_i P^T_{\text{down}}}{R_{s,\text{TST}}}\right) \\ &\quad + + z_{i,s}\left(\frac{\alpha^G_i b_i P^T_{\text{down}}}{R_{s,g}} + \frac{\rho\alpha^G_i b_i P^T_{\text{ISL}}}{R_{\text{ISL}}}\right) \end{aligned} \tag{6}$$

where $z_{i,s}$ is a 0-1 variable to indicate whether satellite s is the nearest one to the ground station on task i's offloading path and $x^G_{i,s}$ represents whether satellite s is on task i's SCC offloading path.

The energy consumption of the cloud center depends on the input data size, given by [1]: $E^{G,\text{com}}_i = A_0\text{exp}(A_1\alpha^G_i b_i)$, where A_0 and A_1 are proportionality coefficients. Compared to satellites, cloud centers have extremely abundant energy supply. Therefore, in the problem formulation described in the next section, we have excluded the cloud center energy consumption.

E. Computation Offloading Cost

Let p_C be the price of satellite computation resources (dollars per CPU cycle). The cost of using satellite computing resources for task i is

$$C^{\text{com}}_i = p_C \sum_{s\in\mathcal{S}} \beta_{i,s}\alpha^S_i b_i h_i. \tag{7}$$

Let $p_{\text{up}}, p_{\text{down}}, p_{\text{ISL}}$ be the prices (dollars per bit) of satellite–terrestrial uplinks, downlinks, and ISLs, respectively. The cost of consuming satellite communication resources for task i is

$$\begin{aligned} C^{\text{trans}}_i = b_i\alpha^S_i\Big[p_{\text{up}} + \sum_{s\in\mathcal{S}}(1+\rho)\beta_{i,s}\text{hop}_{s_0,s}p_{\text{ISL}} + \rho p_{\text{down}}\Big] \\ + (1+\rho)b_i\alpha^G_i\Big[p_{\text{up}} + \text{hop}_{s_0,s_g}p_{\text{ISL}} + p_{\text{down}}\Big] \end{aligned} \tag{8}$$

where the first term is the transmission fee in SEC and the second is that in SCC.

Hence, the total cost of offloading all tasks is

$$C = \sum_{i\in\mathcal{I}} C^{\text{com}}_i + C^{\text{trans}}_i. \tag{9}$$

F. Problem Formulation

Let $\boldsymbol{\alpha}_S = \{\alpha^S_i\}\ \ \forall i \in \mathcal{I}$, and $\boldsymbol{\alpha}_G = \{\alpha^G_i\}\ \ \forall i \in \mathcal{I}$ be the decisions of computation offloading to SEC and SCC, respectively, $\boldsymbol{\beta} = \{\beta_{i,s}\}\ \ \forall i \in \mathcal{I}, s \in \mathcal{S}$ be the proportion of satellite peer offloading in SEC, and $\boldsymbol{\gamma} = \{\gamma_{i,s}\}\ \ \forall i \in \mathcal{I}, s \in \mathcal{S}$ be the decisions of whether a task will be offloaded to a satellite in SEC. The above four sets of variables satisfy the following conditions: $\alpha^S_i + \alpha^G_i = 1$; if $\beta_{i,s} = 0$, then $\gamma_{i,s} = 0$; if $\beta_{i,s} > 0$, then $\gamma_{i,s} = 1$. Based on the pricing policies $\boldsymbol{p}^S = \{p_C, p_{\text{up}}, p_{\text{ISL}}, p_{\text{down}}\}$, the total computation offloading cost is

$$C\Big(\boldsymbol{\alpha}_S, \boldsymbol{\alpha}_G, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{p}^S\Big) = \sum_{i\in\mathcal{I}} C^{\text{com}}_i + C^{\text{trans}}_i. \tag{10}$$

The CE-HCO problem is formulated as

Problem 1:

$$\min_{\boldsymbol{\alpha}_S,\boldsymbol{\alpha}_G,\boldsymbol{\beta},\boldsymbol{\gamma}} \ C\Big(\boldsymbol{\alpha}_S, \boldsymbol{\alpha}_G, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{p}^S\Big) \tag{11}$$

$$\text{s.t.}\ \max\Big\{T^S_i, T^G_i\Big\} \le d_i \quad \forall i \in \mathcal{I} \tag{11a}$$

$$\sum_{i\in\mathcal{I}} \gamma_{i,s} r^s_i \le r^{s,\max} \quad \forall s \in \mathcal{S} \tag{11b}$$

$$\sum_{i\in\mathcal{I}} E^{S,\text{com}}_{i,s} + E^{S,\text{trans}}_{i,s} + E^{G,\text{trans}}_{i,s} \le E^{s,\max} \quad \forall s \in \mathcal{S} \tag{11c}$$

$$\alpha^G_i, \beta_{i,s} \in [0,1], \gamma_{i,s} \in \{0,1\} \quad \forall i \in \mathcal{I}, s \in \mathcal{S}. \tag{11d}$$

Here, $r^{s,\max}$ denotes the maximum computation capacity of satellite s. The left-hand side of (11c) represents the total energy consumption of satellite s serving all tasks, and $E^{s,\max}$ refers to the maximum energy that satellite s can supply. Constraint (11d) defines the allowable values for the variables. In general, Problem 1 aims to minimize the total cost by scheduling the task assignments between SEC and SCC, as well as among satellite peers, under the constraints of deadline requirements and each satellite's computation and energy resources.

V. Problem Solution

In this section, we first utilize the SCA theory and the penalty method to transform the complex CE-HCO problem into a tractable convex problem. Then, we design an algorithm to derive the near-optimal solution. The convergence and complexity of the algorithm are analyzed.

A. Problem Reformulation

Problem 1 is a mixed-integer nonlinear programming (MINLP) problem. In order to reduce the complexity, we first eliminate the $\boldsymbol{\alpha}_G$ in (11) based on $\alpha^S_i + \alpha^G_i = 1$. Then, we relax the discrete variable $\gamma_{i,s}$ into a continuous one, and utilize the penalty method to restrict its value [39], [40]. We add the penalty function $\delta \cdot \gamma_{i,s}(1-\gamma_{i,s})$ to the objective function where the penalty parameter δ takes a large value to amplify the violation of the discrete constraint.

Let $\boldsymbol{\phi} = \{\boldsymbol{\alpha}_S, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$. Thus, by considering all known conditions, we simplify Problem 1 as follows:

Problem 2:

$$\min_{\boldsymbol{\phi}} \sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{S}} \delta\cdot\gamma_{i,s}\left(1-\gamma_{i,s}\right) + \sum_{i\in\mathcal{I}} C_i^1 + \alpha_i^S b_i\left(C_i^2 + \sum_{s\in\mathcal{S}} C_{i,s}^3\beta_{i,s}\right) \tag{12}$$

$$\text{s.t. } C_i^4\alpha_i^S + \max_{s\in\mathcal{S}}\left\{C_{i,s}^5\beta_{i,s}\alpha_i^S + \gamma_{i,s}\cdot\frac{2D_{s_0,s}}{c}\right\} \le d_i - \frac{2D_{\mathrm{TST},s_0}}{c}, \tag{12a}$$

$$\left(1-\alpha_i^S\right)C_i^6 \le d_i - \frac{2\left(D_{\mathrm{TST},s_0} + D_{s_g,g} + D_{s_0,s_g}\right)}{c} \quad \forall i\in\mathcal{I} \tag{12b}$$

$$\sum_{i\in\mathcal{I}} \gamma_{i,s} r_i^s \le r^{s,\max} \quad \forall s\in\mathcal{S} \tag{12c}$$

$$\sum_{i\in\mathcal{I}} \kappa b_i\left(r_i^s\right)^2\beta_{i,s}\alpha_i^S - C_{i,s}^7\alpha_i^S + C_{i,s}^8 \le E^{s,\max} \quad \forall s\in\mathcal{S} \tag{12d}$$

$$\alpha_i^S, \beta_{i,s}, \gamma_{i,s} \in [0,1] \quad \forall i\in\mathcal{I}, s\in\mathcal{S} \tag{12e}$$

where $C_i^1 = (1+\rho)b_i(p_{\mathrm{up}} + \mathrm{hop}_{s_0,s_g}p_{\mathrm{ISL}} + p_{\mathrm{down}})$, $C_i^2 = p_C h_i - \rho p_{\mathrm{up}} - (1+\rho)\mathrm{hop}_{s_0,s_g}p_{\mathrm{ISL}} - p_{\mathrm{down}}$, $C_{i,s}^3 = (1+\rho)\mathrm{hop}_{s_0,s}p_{\mathrm{ISL}}$, $C_i^4 = b_i(1/[R_{\mathrm{TST},s_0}] + \rho/[R_{s_0,\mathrm{TST}}])$, $C_{i,s}^5 = b_i([1/r_i^s] + ([(1+\rho)\mathrm{hop}_{s_0,s}]/R_{\mathrm{ISL}}))$, $C_i^6 = C_i^4 + b_i([1/R_{s_g,g}] + [\rho/R_{g,s_g}] + ([(1+\rho)\mathrm{hop}_{s_0,s_g}]/R_{\mathrm{ISL}}))$, $C_{i,s}^7 = z_{i,s}b_i([P_{\mathrm{down}}^T/R_{s_g,g}] + [\rho P_{\mathrm{ISL}}^T]/R_{\mathrm{ISL}}) + x_{i,s}^G b_i([(1+\rho)P_{\mathrm{ISL}}^T]/R_{\mathrm{ISL}}) - x_{i,s}^S b_i([(1+\rho)P_{\mathrm{ISL}}^T]/R_{\mathrm{ISL}})$, $C_{i,s}^8 = b_i[x_{i,s}^G([(1+\rho)P_{\mathrm{ISL}}^T]/R_{\mathrm{ISL}}) + y_{i,s}([P_{\mathrm{ISL}}^T/R_{\mathrm{ISL}}] + [\rho P_{\mathrm{down}}^T/R_{s_0,\mathrm{TST}}]) + z_{i,s}([P_{\mathrm{down}}^T/R_{s_g,g}] + [\rho P_{\mathrm{ISL}}^T/R_{\mathrm{ISL}}])]$ are constants. Constraint (11a) is split into two constraints: 1) $T_i^S \le d_i$ (12a) and 2) $T_i^G \le d_i$ (12b).

It is clear that Problem 2 is nonconvex, primarily due to the product of two decision variables, $\alpha_i^S\beta_{i,s}$, in the objective function and constraints. Additionally, the term $\gamma_{i,s}(1-\gamma_{i,s})$ is concave.

Problem 2 is NP-hard due to its nonconvexity, which is further exacerbated by the scale of the network. To address this, we employ the parallel SCA algorithm [41], which is capable of handling multivariable nonconvex problems without requiring the convexity of the objective function and constraints [42]. With various approximation options available, the parallel SCA algorithm provides the necessary flexibility to transform the nonconvex optimization problem into an approximate convex problem, converging after a limited number of iterations [43]. Using the SCA theory, we transform Problem 2 into a tractable convex problem.

Theorem 1: Based on the guidance on choosing surrogate function in [43], we can deduce that when V is a differentiable concave function on a feasible set X, the convex surrogate of V can be its first order Taylor expansion on $\mathbf{x}^k \in X$: $\widetilde{V}(\mathbf{x}|\mathbf{x}^k) = V(\mathbf{x}^k) + \nabla V(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k)$.

Theorem 2: From the parallel implementation example in [43], we can deduce that when $\mathbf{x}$ consists of n blocks, i.e., $\mathbf{x} = [\mathbf{x}_i^T, \ldots, \mathbf{x}_n^T]^T$ and each $\mathbf{x}_i \in \mathbb{R}^{m_i}$, the surrogate function $\widetilde{V}$ is additively separable in the blocks, i.e., $\widetilde{V}(\mathbf{x}|\mathbf{x}^k) = \sum_{i=1}^n \widetilde{V}_i(\mathbf{x}_i|\mathbf{x}^k)$.

According to Theorems 1 and 2, for the given feasible solution $\boldsymbol{\phi}^{(k)} = \{\boldsymbol{\alpha}_S^{(k)}, \boldsymbol{\beta}^{(k)}, \boldsymbol{\gamma}^{(k)}\}$ at the kth iteration of the SCA algorithm, the convex approximation of $\sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{S}}\delta\cdot\gamma_{i,s}(1-\gamma_{i,s})$ can be

$$M_1(\boldsymbol{\gamma}) = \delta\sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{S}}\gamma_{i,s}^{(k)}\left(1-\gamma_{i,s}^{(k)}\right) + \left(-2\gamma_{i,s}^{(k)}+1\right)\left(\gamma_{i,s}-\gamma_{i,s}^{(k)}\right). \tag{13}$$

Theorem 3: According to the choice of surrogate functions in [43], if function V is written as the product of functions, i.e., $V(\mathbf{x}) = V_1(\mathbf{x})V_2(\mathbf{x})$, where V_1 and V_2 are convex and nonnegative on X and $\mathbf{x}^k \in X$, the convex approximation of function V can be expressed as follows: $\widetilde{V}(\mathbf{x}|\mathbf{x}^k) = V_1(\mathbf{x})V_2(\mathbf{x}^k) + V_1(\mathbf{x}^k)V_2(\mathbf{x}) + [\boldsymbol{\tau}/2](\mathbf{x}-\mathbf{x}^k)^T\mathbf{H}(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k)$, where $\boldsymbol{\tau}$ can be any positive constant matrix and $\mathbf{H}$ can be any positive definite matrix.

Based on Theorem 3, the convex approximation of $\alpha_i^S\beta_{i,s}$ can be obtained by

$$M_2\left(\alpha_i^S, \beta_{i,s}\right) = \alpha_i^S\beta_{i,s}^{(k)} + \alpha_i^{S(k)}\beta_{i,s} + \frac{\tau_\alpha}{2}\left(\alpha_i^S - \alpha_i^{S(k)}\right)^2 + \frac{\tau_\beta}{2}\left(\beta_{i,s} - \beta_{i,s}^{(k)}\right)^2. \tag{14}$$

Therefore, the convex approximation problem of Problem 2 at kth iteration can be given by

Problem 2':

$$\min_{\boldsymbol{\phi}} M_1(\boldsymbol{\gamma}) + \sum_{i\in\mathcal{I}} C_i^1 + \alpha_i^S b_i C_i^2 + \sum_{s\in\mathcal{S}} b_i C_{i,s}^3 M_2\left(\alpha_i^S, \beta_{i,s}\right) \tag{15}$$

$$\text{s.t. } C_i^4\alpha_i^S + C_{i,s}^5 M_2\left(\alpha_i^S, \beta_{i,s}\right) + \gamma_{i,s}\cdot\frac{2D_{s_0,s}}{c} \le d_i - \frac{2D_{i,s_0}}{c} \quad \forall i\in\mathcal{I}, s\in\mathcal{S} \tag{15a}$$

$$\sum_{i\in\mathcal{I}}\kappa b_i\left(r_i^s\right)^2 M_2\left(\alpha_i^S, \beta_{i,s}\right) - C_{i,s}^7\alpha_i^S + C_{i,s}^8 \le E^{s,\max} \quad \forall s\in\mathcal{S},$$

$$\text{(12b), (12c), (12e)} \tag{15b}$$

where (12a) is split into multiple constraints (15a), each under a pair of i and s.

B. Algorithm to Solve Problem 2′

To solve Problem 2′, we approach its solution iteratively until a convergence condition is met or the total number of iteration rounds surpasses the predefined maximum. We define $\boldsymbol{\varepsilon} = \{\varepsilon_i\}\ \forall i\in\mathcal{I}$ as a threshold. The convergence condition is satisfied if the following expression holds for any task i:

$$\varepsilon_i^{(k)} = \frac{\left\|\alpha_i^{S(k)} - \alpha_i^{S(k-1)}\right\|}{\left\|\alpha_i^{S(k)}\right\|} + \frac{\left\|\boldsymbol{\beta}_i^{(k)} - \boldsymbol{\beta}_i^{(k-1)}\right\|}{\left\|\boldsymbol{\beta}_i^{(k)}\right\|} + \frac{\left\|\boldsymbol{\gamma}_i^{(k)} - \boldsymbol{\gamma}_i^{(k-1)}\right\|}{\left\|\boldsymbol{\gamma}_i^{(k)}\right\|} \le \varepsilon_i. \tag{16}$$

```
Algorithm 1 Algorithm to Solve the CE-HCO
Input: Maximum iteration number $K$ and threshold $\boldsymbol{\varepsilon}$.
Output: Offloading decision $\boldsymbol{\phi}$ and computation offloading cost $C$.
Initialize $\boldsymbol{\alpha}_S = 0$, $\boldsymbol{\phi} = \boldsymbol{\phi}^{(0)}$, $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{(0)}$, $k = 0$, $v^0 = 1$;
repeat
    Solve $P2'$ and obtain optimal solution $\left\{\boldsymbol{\phi}^*(\boldsymbol{\phi}^{(k)}), R^{*(k)}\right\}$;
    Update $\boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)} + v^k(\boldsymbol{\phi}^*(\boldsymbol{\phi}^{(k)}) - \boldsymbol{\phi}^{(k)})$;
    Update $v^{k+1} = \frac{v^k + w_1(k)}{1 + w_2(k)}$;
    $k = k + 1$;
until $\forall i \in \mathcal{I}, \varepsilon_i^{(k)} \le \varepsilon_i$ or $k > K$
```

To choose appropriate iteration step-sizes, we use a diminishing step-size rule where $v^k = ([v^{k-1} + w_1(k)]/[1 + w_2(k)])$ with $k = 1, \ldots,$ and $v^0 = 1$. The $w_1(k)$ and $w_2(k)$ are two nonnegative real functions of $k \geq 1$ such that: $0 \leq w_1(k) \leq w_2(k)$, $w_1(k)/w_2(k) \rightarrow 0$ as $k \rightarrow \infty$, and $\sum_k (w_1(k)/w_2(k)) = \infty$. Here, we take $w_1(k)$ and $w_2(k)$ as suggested in [43] suggests: $w_1(k) = w_1$ and $w_2(k) = w_2 \cdot k$, where w_1 and w_2 are given constants satisfying $w_1, w_2 \in (0, 1)$ and $w_1 \leq w_2$.

The algorithm to derive the solution of Problem $2'$ is presented in Algorithm 1.

C. Analysis of the Algorithm

Theorem 4: The algorithm converges to the optimal solution of Problem $2'$.

Proof: Based on [43], if the objective function $V(\mathbf{x}|\mathbf{x}^k)$ is strongly convex and differentiable on its domain X, the parallel SCA algorithm will converge to a stationary point with the variable update rule $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + v^k(\mathbf{x}^*(\mathbf{x}^{(k)}) - \mathbf{x}^{(k)})$ and a diminishing step-size rule. Specifically, as k goes to infinity, the norm of the difference between the optimal solution $\mathbf{x}^*(\mathbf{x}^k)$ and the current iterate $\mathbf{x}^k$ approaches zero, i.e., $\lim_{x \to \infty} \|\mathbf{x}^*(\mathbf{x}^k) - \mathbf{x}^k\| = 0$.

Theorem 5: The computation complexity of the algorithm is $\mathcal{O}(K \cdot (3IS)^3)$.

Proof: According to [43], in each iteration, the values of three valuables, $\alpha_i^{S(k)}$, $\boldsymbol{\beta}_i^{(k)}$, and $\boldsymbol{\gamma}_i^{(k)}$, associated with every task i and every satellite s, are updated. As a result, the computation complexity for each iteration is $\mathcal{O}((3IS)^3)$. Given that the maximum number of iterations is K, the computation complexity of the algorithm is $\mathcal{O}(K \cdot (3IS)^3)$.

VI. Simulations

In this section, we verify the effectiveness of our work through extensive simulations. First, we demonstrate the convergence and optimality of the proposed algorithm. Next, we analyze the performance of CE-HCO under several task settings and pricing policies.

A. Simulation Settings

We adopt a satellite constellation of 66 satellites, with 11 orbit planes each containing 6 satellites. The constellation altitude is 570 km, and the inclination is 70°, which is the same as the Starlink constellation, phase 1, group 2. We employ the Satellite Tool Kit to derive the longitude and latitude of each satellite, and the distance between any two satellites at any given simulation instant. The propagation latency for these links is calculated by dividing the intersatellite distances by the speed of light. The user elevation angle is set at 10°, indicating that UEs are capable of accessing satellites within an elevation range of 10° to 90° from the horizontal plane. Each satellite has a computation capacity of 2 Gcycles/s [24] and can manage at most 6 resource blocks in one slot [46]. We assume that the computation capacity is equally distributed among the resource blocks. The transmission rate for uplink and downlink is 20 and 150 Mb/s, respectively, which is the same as Starlink's current average service rate [47]. We assume that the data rates for all uplinks and downlinks are the same in the simulation for simplicity, since our scheme focuses on computation offloading scheduling and routing in STIN. Similar simplifications are adopted in [10] for clarity of the simulations. According to [44], the maximum power output of the LEO satellite's solar panels is 500W. Therefore, we assume that 100W is reserved for computation offloading on each satellite.

The ground cloud center is located at (34.05N, 118.24W) in Los Angeles, where the AWS's cloud computing infrastructure is deployed. We have placed four sets of IoT devices at (38.5N, 46.5E), (58.5S, 145.5E), (41.5S, 14.5E), and (46.5S, 81.5E), respectively. These devices are 7130, 8510, 9930, and 11200-km away from the cloud center, respectively. Each set consists of 100 devices, and each device generates one task in each time slot.

The length of a satellite constellation's snapshot is 60 s [48], which is much larger than the tasks' deadlines. Hence, we schedule computation offloading under one fixed snapshot.

We have adopted the satellite operator's communication resource charge policy, which charges approximately \$1 for transferring 1 GB of data, based on ViaSAT's pricing policy. This is ten times of the ground bandwidth charge, which is \$0.1 for transferring 1 GB of data by Amazon CloudFront [32]. According to [14], a rough estimate suggests that the service cost of a satellite server is at least three times higher than a cloud center server. Therefore, we assume that the satellite operator's computation resource charge policy is \$3.6/h, which is eight times of the existing cloud computing charge policies, such as Amazon EC2 m5 and large instance. Other simulation parameters are described in Table III.

B. Convergence and Optimality of the Proposed Algorithm

We compare the performance of the proposed algorithm with the optimal solution and one of the commonly used multiobjective optimization heuristic algorithms, i.e., the multiobjective particle swarm optimization (MOPSO) algorithm. The results were averaged over 100 random realizations.

As shown in Fig. 4, the proposed algorithm reaches its optimal solution in the 6th iteration, while MOSPO converges until the 16th iteration. Since our continuous convex approximation method takes the lower limit of the real function value

TABLE III
SIMULATION PARAMETERS

Parameter	Value
κ	10^{-26} [24]
R_{ISL}	1Gbps [44]
P_{ISL}^{T}, P_{down}^{T}	50W [44], 2W [45]
ρ	0.1 [25]
b_i	0.5-3 Mbit [24]
h_i	120 [24]
d_i	0.6-2.35s
r_i^L	0.1Gcycles/s [34]
E_i	100W

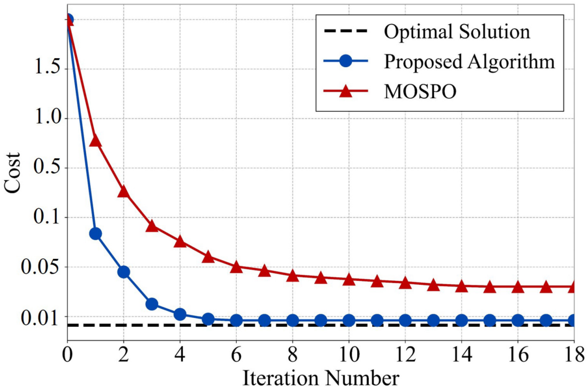


Fig. 4. Convergence and optimality of the proposed algorithm.

at each point, the proposed algorithm cannot reach the optimal value. However, it is close enough to the optimal solution. In contrast, the MOSPO algorithm is far from the optimal solution because, in the case of a large solution space, it is easy to fall into a local optimum.

C. Superiority of Hybrid Offloading

We incorporate two baseline offloading schemes for comparative analysis. First, we refer to the satellite peer offloading scheme detailed in [11] as the SEC-only scheme, which offloads tasks to satellite nodes with lighter burdens within specified deadlines. Additionally, we adopt the pure SCC scheme, offloading tasks to cloud centers by adhering to satellite energy limitations and meeting required deadlines in the most economical manner.

The results illustrated in Figs. 5 and 6 reveal that the success ratios for our proposed hybrid scheme and the SEC-only approach are identical, both consistently outperforming the SCC-only model. This outcome is attributed to the SEC's advantage in reducing propagation delays compared to the SCC. Notably, the success ratio increases as latency requirements become more lenient, allowing for a higher number of tasks to be completed within their deadlines. Although the success rates for the hybrid and SEC-only schemes are similar, our hybrid approach achieves greater cost reductions as latency constraints are relaxed. This is because more tasks can be economically offloaded to the SCC within their deadlines, leveraging cost-effective terrestrial computing resources. The costs associated with the SCC-only and SEC-only escalate with more lenient latency settings due to an increase in the number of tasks successfully completed.

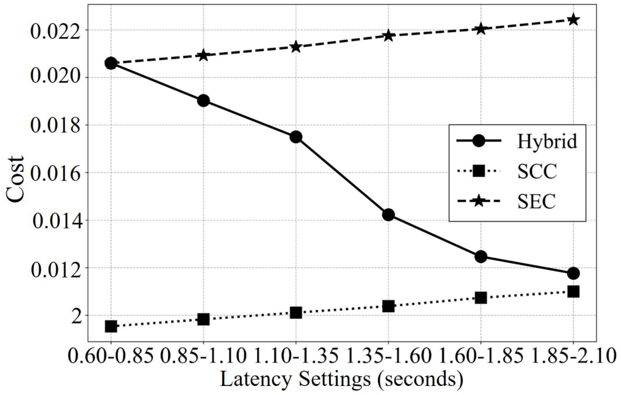


Fig. 5. Cost of computation offloading, varying with the latency settings and offloading scheme (device set 3 with fixed 3-Mb task size and pricing policy).

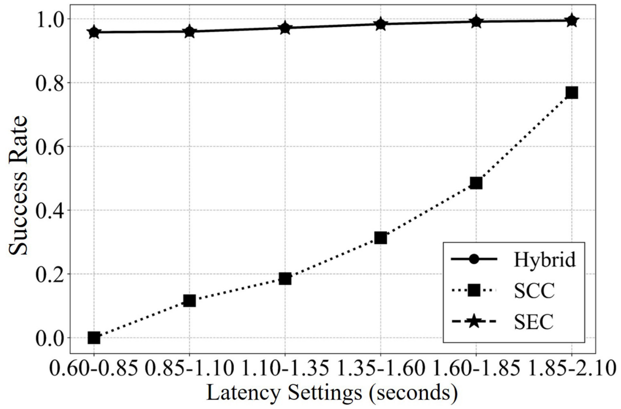


Fig. 6. Ratio of successfully offloaded tasks, varying with the latency settings and offloading scheme (device set 3 with fixed 3-Mb task size and pricing policy).

In summary, our proposed hybrid offloading scheme not only achieves the optimal success rate but also minimizes offloading costs, evidencing its superiority over the baseline strategies.

D. Cost-Effective Computation Offloading Scheduling

In this section, we undertake a series of simulations aimed at uncovering the key factors related to computation offloading decisions.

First, we investigate how the latency requirements of tasks influence these offloading decisions.

As shown in Fig. 7, the task input size is fixed and the pricing policy is predetermined: satellite communication resources are priced at a rate tenfold that of terrestrial networks, and satellite computational resources are priced eightfold. We consider four sets of IoT devices, located at 7130, 8510, 9930, and 11200 km from the cloud center, respectively. From Fig. 7, it is apparent that when the distance remains constant, tasks with urgent latency demands are offloaded to the SEC with greater frequency due to satellite's closer

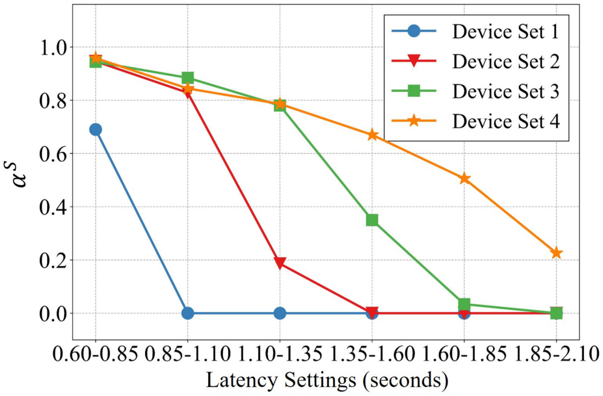


Fig. 7. Ratio of tasks offloaded to SEC, varying with the latency settings and distance to the cloud center (with fixed 3-Mb task size and pricing policy).

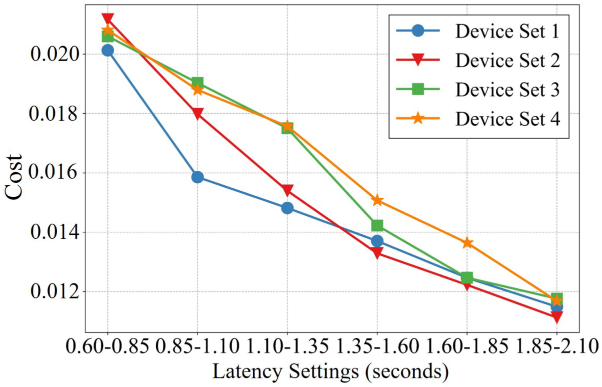


Fig. 8. Cost of computation offloading, varying with the latency settings and distance to the cloud center (with fixed 3-Mb task size and pricing policy).

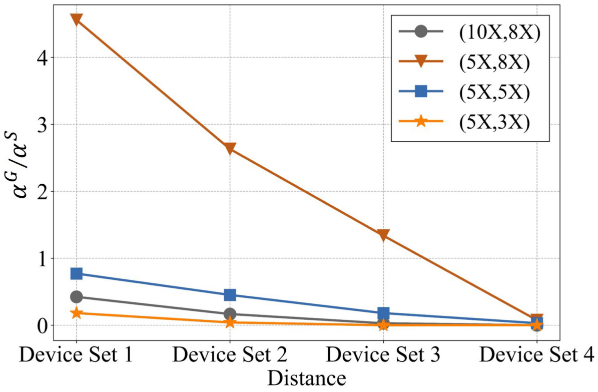


Fig. 9. Ratio of α^G/α^S, varying with distance to the cloud center and pricing policies (with fixed 3-Mb task size and 2.1–2.35-s latency requirement).

proximity to devices at an altitude of 570 km. As the latency requirement relaxes, fewer tasks are offloaded to SEC from an economic perspective. On the other hand, keeping the deadline requirements consistent, we find that devices positioned at greater distances from the cloud center tend to offload tasks to the SEC more frequently. This pattern emerges as the longer propagation round-trip-time to the cloud center cannot guarantee the latency requirements. It is important to note that the distances referred to herein represent the direct, straight-line distances on a map. However, the actual transmission paths within the satellite network are not linear but rather follow a zigzagged trajectory, resulting in longer effective distances. By 'zigzagged,' it is meant that the path involves multiple intersatellite hops, forming a broken line rather than a straight path. Consequently, the actual disparity in distance for different sets of devices is more significant than the straight-line measurements would suggest. In Fig. 8, the offloading cost decreases as more tasks are offloaded to the cloud center. This is because the price of SEC is higher than SCC.

The analyses of Figs. 7 and 8 yield valuable lessons regarding the economic strategy for satellite computing services. Given the costs associated with utilizing satellite communication and computational resources, cloud providers and satellite operators have the opportunity to adjust the service fees charged to users based on specific task parameters. For tasks that demand urgency and originate from devices at extended distances, it would be judicious to increase the service fees within in a reasonable range. This would appropriately compensate the higher SEC cost, ensuring a sustainable service model.

Next, we investigate how the computation offloading decisions respond to different pricing policies to minimize the cost. In this case, we consider four pricing policies of satellite communication and computation resources. According to the current satellite communication pricing policy of ViaSAT, transferring 1GB data costs $1, which is 10 times more than the ground [32]. Hence, we adopt this as the current satellite communication resource price. Meanwhile, with lower weights and smaller sizes, LEO satellites enable cheaper design, easy mass production, and low-cost launching. It is reasonable to believe that the price of satellite communication will be reduced in the near future, so we take the price of 5 times of the ground as the future satellite communication price in our simulations. As for the satellite computation resource, Bhattacherjee et al. [14] estimated that the price is at least three times of the ground cloud center server. Thus, we set the price of satellite computation resources to be 8 times, 5 times, and 3 times of the ground, which decreases with the progress of satellite chip manufacturing technology. In summary, we consider four pricing policies in the long run: (10×, 8×) (which means the satellite communication price is 10 times of ground and the computation price is 8 times), and (5×, 8×), (5×, 5×), and (5×, 3×). There is no such setting as (10×, 5×) here since it has been observed that communication prices tend to drop before computing prices. This trend is exemplified by ViaSAT, a satellite broadband service provider whose costs have decreased by two-thirds since 2017 [49]. However, the availability of SEC to the general public is still pending.

Fig. 9 demonstrates that α^G/α^S reduces with the decreased ratio of satellite computation and communication resource price for a given device set. It means that the more computing resources expensive than communication resources, more tasks are forwarded to the cloud center for computing via the cost-effective satellite communication links. Conversely, if computation is cheap, it would be better to process tasks directly on satellites rather than forwarding over a long

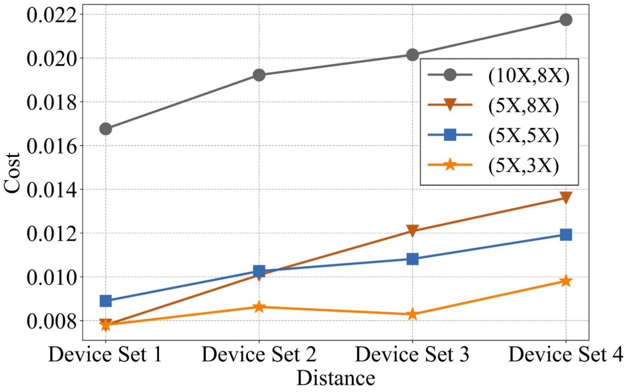


Fig. 10. Computation offloading cost, varying with distance to the cloud center and pricing policies (with fixed 3-Mb task size and 2.1–2.35-s latency requirement).

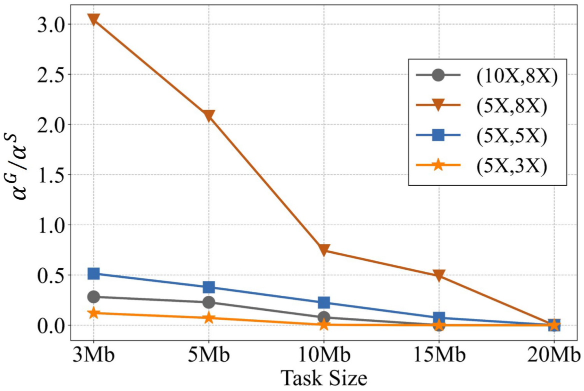


Fig. 11. Ratio of α^G/α^S, varying with task sizes and pricing policies (device set 1 with 2.1–2.35-s latency requirement).

distance to save cost and delay. Moreover, α^G/α^S decreases with the increase in devices' distances to the cloud center, because the cost benefits of satellite computing become more pronounced over multihop forwarding when the distance is extended. As depicted in Fig. 10, although computation offloading decisions alter their offloading ratios in response to pricing policies, there still remains a general trend where an increase in the pricing correlates with a rise in overall costs. Lastly, given the observed trend where costs escalate with increasing distance, it can be deduced that transmission expenses comprise a substantial fraction of the overall cost. In Fig. 11, it is observed that an increase in task size corresponds to a decrease in the ratio of offloading to SCC, which ultimately drops to zero. This trend is attributable to the escalating latency from multihop transmission to the cloud center, which eventually surpasses the specified latency constraints. As a result, to adhere to the latency requirements, tasks must either be processed directly on satellites or they risk failure due to processing delays.

From Fig. 9 to 11, the insights gained highlight that cloud providers and satellite operators should consider not only task parameters but also satellite resource pricing policies when devising strategies for service fee charges. These policies should reflect the capacities of on-board satellite resources as well as the overheads associated with maintaining and updating the satellite system. By integrating these considerations, providers can promote a sustainable expansion of satellite computing services.

VII. Conclusion

This article have introduced a novel CE-HCO paradigm within STINs, enabling public cloud providers and satellite operators to offer pay-as-you-go computing services anytime and anywhere. We have presented a comprehensive outline of the CE-HCO framework and its workflow. Under this framework, we have formulated a CE-HCO optimization problem aimed at minimizing computation offloading costs while adhering to user-defined latency requirements and satellite energy limitations. To address the inherent complexity of this MINLP problem, we have applied SCA theory and the penalty method to reformulate the problem into a tractable, convex one. The proposed algorithm has delivered near-optimal solutions with satisfactory convergence rates and computational complexity. Through simulation studies, we have demonstrated how variables, such as task latency requirements, task sizes, and device proximity, to cloud centers, and satellite resource pricing policies influence CE-HCO decisions. Furthermore, the simulations have offered strategic insights for public cloud providers and satellite operators on structuring service fees for satellite computing services to optimize long-term cost efficiency.

References

[1] Q. Chen, W. Meng, T. Q. S. Quek, and S. Chen, "Multi-tier hybrid offloading for computation-aware IoT applications in civil aircraft-augmented SAGIN," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 399–417, Feb. 2023.

[2] S. Kavuri, D. Moltchanov, A. Ometov, S. Andreev, and Y. Koucheryavy, "Performance analysis of onshore NB-IoT for container tracking during near-the-shore vessel navigation," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2928–2943, Apr. 2020.

[3] R. Lou, Z. Lv, S. Dang, T. Su, and X. Li, "Application of machine learning in ocean data," *Multimedia Syst.*, vol. 29, pp. 1815–1824, Jun. 2023.

[4] J. Xie, S. Liu, and X. Wang, "Framework for a closed-loop cooperative human cyber-physical system for the mining industry driven by VR and AR: MHCPS," *Comput. Ind. Eng.*, vol. 168, Jun. 2022, Art. no. 108050.

[5] J. Novet. "Google wins cloud deal from Elon musk's SpaceX for Starlink Internet connectivity." Accessed: Nov. 10, 2023. [Online]. Available: https://www.cnbc.com/2021/05/13/google-cloud-wins-spacex-deal-for-starlink-Internet-connectivity.html

[6] G. Reim, (Aviation Week Netw. Co., New York, NY, USA). *Why Amazon Web Services is Going to Space*. Accessed: Nov. 10, 2023. [Online]. Available: https://aviationweek.com/aerospace/commercial-space/why-amazon-web-services-going-space

[7] M. Sheetz. "Amazon used AWS on a satellite in orbit to speed up data analysis in "first-of-its kind" experiment." 2022. [Online]. Available: https://www.cnbc.com/2022/11/29/amazon-aws-experiment-satellite-orbit.html

[8] "Study on integration of satellite components in the 5G architecture phase III," 3GPP, Sophia Antipolis, France, document SP-231199, 2023.

[9] "Study on 5G system with satellite backhaul," 3GPP, Sophia Antipolis, France, Rep. TR-23.700, 2022.

[10] N. Cheng et al., "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[11] X. Zhang et al., "Energy-efficient computation peer offloading in satellite edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 3077–3091, Apr. 2024.

[12] X. Cao et al., "Edge-assisted multi-layer offloading optimization of LEO satellite–terrestrial integrated networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 381–398, Feb. 2023.

[13] X. Zhu and C. Jiang, "Delay optimization for cooperative multi-tier computing in integrated satellite–terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 366–380, Feb. 2023.

[14] D. Bhattacherjee, S. Kassing, M. Licciardello, and A. Singla, "In-orbit computing: An outlandish thought experiment?" in *Proc. 19th ACM Workshop HotNets*, 2020, pp. 197–204.

[15] L. Yan et al., "SatEC: A 5G satellite edge computing framework based on microservice architecture," *Sensors*, vol. 19, no. 4, p. 831, 2019.

[16] M. El Jaafari, N. Chuberre, S. Anjuere, and L. Combelles, "Introduction to the 3GPP-defined NTN standard: A comprehensive view on the 3GPP work on NTN," *Int. J. Satell. Commun. Netw.*, vol. 41, no. 3, pp. 220–238, 2023.

[17] L. Bernstein, (Kratos Def. Secur. Solut. IT Secur. Co., San Diego, CA, USA). *Satellite Industry Leaders Discuss Cloud Adoption, Evolution of Edge Computing*. 2022. [Online]. Available: https://www.kratosdefense.com/constellations/articles/satellite-industry-leaders-discuss-cloud-adoption-evolution-of-edge-computing

[18] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting Internet of remote Things," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 113–123, Feb. 2016.

[19] W.-C. Chien, C.-F. Lai, M. S. Hossain, and G. Muhammad, "Heterogeneous space and terrestrial integrated networks for IoT: Architecture and challenges," *IEEE Netw.*, vol. 33, no. 1, pp. 15–21, Jan./Feb. 2019.

[20] D. Swinhoe, (DatacenterDynamics Com., London, U.K.). *Loft Orbital to Work With US Space Force on Satellite "Edge Computing" Capabilities*. 2021. [Online]. Available: https://www.datacenterdynamics.com/en/news/loft-orbital-to-work-with-us-space-force-on-satellite-edge-computing-capabilities/

[21] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Satellite–terrestrial integrated edge computing networks: Architecture, challenges, and open issues," *IEEE Netw.*, vol. 34, no. 3, pp. 224–231, May/Jun. 2020.

[22] S. Cao, H. Han, J. Wei, Y. Zhao, S. Yang, and L. Yan, "Space cloud-fog computing: Architecture, application and challenge," in *Proc. 3rd CSAE*, 2019, pp. 1–7.

[23] Z. Yu, X. Feng, T. Dai, and Z. Lu, "Space edge computing: Requirement, architecture and key technique," *J. Electron. Inf. Technol.*, vol. 44, no. 12, pp. 4416–4425, 2022.

[24] Z. Song, Y. Hao, Y. Liu, and X. Sun, "Energy-efficient multiaccess edge computing for terrestrial-satellite Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 14202–14218, Sep. 2021.

[25] F. Chai, Q. Zhang, H. Yao, X. Xin, R. Gao, and M. Guizani, "Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7783–7795, Jun. 2023.

[26] J. Du, Y. Sun, N. Zhang, Z. Xiong, A. Sun, and Z. Ding, "Cost-effective task offloading in NOMA-enabled vehicular mobile edge computing," *IEEE Syst. J.*, vol. 17, no. 1, pp. 928–939, Mar. 2023.

[27] X. Wang, J. Wang, X. Zhang, X. Chen, and P. Zhou, "Joint task offloading and payment determination for mobile edge computing: A stable matching based approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12148–12161, Oct. 2020.

[28] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 1975–1989, Sep. 2019.

[29] Q. Wang, S. Guo, J. Liu, C. Pan, and L. Yang, "Profit Maximization incentive mechanism for resource providers in mobile edge computing," *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 138–149, Jan./Feb. 2022.

[30] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: Improving QoS of high-speed satellite–terrestrial networks using edge computing techniques," *IEEE Netw.*, vol. 33, no. 1, pp. 70–76, Jan./Feb. 2019.

[31] Y. Li et al., "A case for stateless mobile core network functions in space," in *Proc. ACM SIGCOMM*, 2022, pp. 298–313.

[32] Z. Lai, H. Li, Q. Zhang, Q. Wu, and J. Wu, "Cooperatively constructing cost-effective content distribution networks upon emerging low Earth orbit satellites and clouds," in *Proc. 29th ICNP*, 2021, pp. 1–12.

[33] T. Zhang, H. Li, S. Zhang, J. Li, and H. Shen, "STAG-based QoS support routing strategy for multiple missions over the satellite networks," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6912–6924, Oct. 2019.

[34] Q. Tang, Z. Fei, B. Li, and Z. Han, "Computation offloading in LEO satellite networks with hybrid cloud and edge computing," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9164–9176, Jun. 2021.

[35] R. Deng, B. Di, S. Chen, S. Sun, and L. Song, "Ultra-dense LEO satellite offloading for terrestrial networks: How much to pay the satellite operator?" *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6240–6254, Oct. 2020.

[36] M. Siew, K. Guo, D. Cai, L. Li, and T. Q. Quek, "Let's share VMs: Optimal placement and pricing across base stations in MEC systems," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2021, pp. 1–10.

[37] Z. Tao et al., "A survey of virtual machine management in edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1482–1499, Aug. 2019.

[38] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, Oct. 2018.

[39] Q.-V. Pham, S. Mirjalili, N. Kumar, M. Alazab, and W.-J. Hwang, "Whale optimization algorithm with applications to resource allocation in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4285–4297, Apr. 2020.

[40] A. Khalili, S. Akhlaghi, H. Tabassum, and D. W. K. Ng, "Joint user association and resource allocation in the uplink of heterogeneous networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 6, pp. 804–808, Jun. 2020.

[41] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization—Part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, Apr. 2017.

[42] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[43] A. Nedić, J.-S. Pang, G. Scutari, Y. Sun, G. Scutari, and Y. Sun, "Parallel and distributed successive convex approximation methods for big-data optimization," in *Multi-agent Optimization*. Cetraro, Italy: Springer, 2018, pp. 141–308.

[44] Y. Yang, M. Xu, D. Wang, and Y. Wang, "Towards energy-efficient routing in satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3869–3886, Dec. 2016.

[45] Y. Liu, Y. Wang, J. Wang, L. You, W. Wang, and X. Gao, "Robust downlink precoding for LEO satellite systems with per-antenna power constraints," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10694–10711, Oct. 2022.

[46] Y. Li, X. Wang, X. Gan, H. Jin, L. Fu, and X. Wang, "Learning-aided computation offloading for trusted collaborative mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 12, pp. 2833–2849, Dec. 2020.

[47] J. Brodkin. "Starlink is getting a lot slower as more people use it, speed tests show." 2022. [Online]. Available: https://arstechnica.com/tech-policy/2022/09/ookla-starlinks-median-us-download-speed-fell-nearly-30mbps-in-q2-2022/

[48] I. Akyildiz, E. Ekici, and M. Bender, "MLSR: A novel routing algorithm for multilayered satellite IP networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 411–424, Jun. 2002.

[49] A. Harebottle. "Bandwidth pricing: How low can it go?" Accessed: Nov. 10, 2023. [Online]. Available: https://interactive.satellitetoday.com/bandwidth-pricing-how-low-can-it-go/

Xinyuan Zhang (Graduate Student Member, IEEE) received the B.S. degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2019, and the Ph.D. degree from the State Key Laboratory of Networking and Switching Technology, BUPT in 2024.

She was a visiting Ph.D. student with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore, from 2022. Her current research interests include satellite–terrestrial integrated networks, edge intelligence, and generative AI.

Jiang Liu received the B.S. degree in electronics engineering from Beijing Institute of Technology, Beijing, China, in 2005, the M.S. degree in communication and information system from Zhengzhou University, Zhengzhou, China, in 2009, and the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, in 2012.

He is currently a Professor with Beijing University of Posts and Telecommunications and also with the Future Network Research Center, Purple Mountain Laboratories, Nanjing, China. His current research interests include network architecture, network virtualization, satellite networking, software-defined networking, information-centric networking, and network testbed.

Zehui Xiong (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2019.

He is currently an Assistant Professor with Singapore University of Technology and Design, Singapore, and also an Honorary Adjunct Senior Research Scientist with Alibaba-NTU Singapore Joint Research Institute, Singapore. He was the Visiting Scholar with Princeton University, Princeton, NJ, USA, and the University of Waterloo, Waterloo, ON, Canada. He has published more than 150 research papers in leading journals and flagship conferences and many of them are ESI highly cited papers. His research interests include wireless communications, Internet of Things, blockchain, edge intelligence, and metaverse.

Dr. Xiong has won over ten Best Paper Awards in international conferences and is listed in the World's Top 2% Scientists identified by Stanford University. He is the recipient of the IEEE Early Career Researcher Award for Excellence in Scalable Computing, the IEEE Technical Committee on Blockchain and Distributed Ledger Technologies Early Career Award, the IEEE Internet Technical Committee Early Achievement Award, the IEEE Best Land Transportation Paper Award, the IEEE CSIM Technical Committee Best Journal Paper Award, the IEEE SPCC Technical Committee Best Paper Award, the IEEE VTS Singapore Best Paper Award, the Chinese Government Award for Outstanding Students Abroad, and the NTU SCSE Best PhD Thesis Runner-Up Award. He is currently serving as the Editor or Guest Editor for many leading journals, including IEEE Journal on Selected Areas in Communications, IEEE Transactions on Vehicular Technology, IEEE Internet of Things Journal, IEEE Transactions on Cognitive Communications and Networking, *IEEE Transactions on Network Science and Engineering*, IEEE Systems Journal, and IEEE/CAA Journal of Automatica Sinica. He is currently serving as the Associate Director for Future Communications Research and Development Programme.

Yudong Huang (Graduate Student Member, IEEE) received the B.S. degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2019, and the Ph.D. degree from the State Key Laboratory of Networking and Switching Technology, BUPT in 2024.

He was a visiting Ph.D. student with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, from 2022. His current research interests include time-sensitive networks, deterministic networks, and network architecture.

Ran Zhang (Member, IEEE) received the Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2021.

He is currently an Associate Research Fellow with the State Key Laboratory of Networking and Switching Technology, BUPT and also with Future Network Research Center, Purple Mountain Laboratories, Nanjing, China. His research interests include satellite–terrestrial integrated networks and artificial intelligence.

Shiwen Mao (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA, in 2004.

He is a Professor, the Earle C. Williams Eminent Scholar, and the Director of the Wireless Engineering Research and Education Center, Auburn University, Auburn, AL, USA. His research interest includes wireless networks, multimedia communications, and smart grid.

Prof. Mao received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019 and the NSF CAREER Award in 2010. He is a co-recipient of the 2021 Best Paper Award of Elsevier/KeAi Digital Communications and Networks Journal, the 2021 IEEE Internet of Things Journal Best Paper Award, the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the 2018 Best Journal Paper Award, the 2017 Best Conference Paper Award from IEEE ComSoc MMTC, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a co-recipient of the Best Paper Awards from IEEE ICC 2022 and 2013; IEEE GLOBECOM 2019, 2016, and 2015; and IEEE WCNC 2015, and the Best Demo Awards from IEEE INFOCOM 2022 and IEEE SECON 2017.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland at College Park, College Park, MD, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland at College Park. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department as well as the Computer Science Department, University of Houston, Houston, TX, USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea. His main research targets on the novel game-theory related concepts critical to enabling efficient and distributive use of wireless networks with limited resources. His other research interests include wireless resource allocation and management, wireless communications and networking, quantum computing, data science, smart grid, carbon neutralization, security, and privacy.

Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JSAC) in 2016, IEEE Vehicular Technology Society 2022 Best Land Transportation Paper Award, and several best paper awards in IEEE conferences. He is a 1% Highly Cited Researcher since 2017 according to Web of Science. He is also the Winner of the 2021 IEEE Kiyo Tomiyasu Award (an IEEE Field Award), for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks." He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018 and an ACM Distinguished Speaker from 2022 to 2025, an AAAS Fellow since 2019, and an ACM Fellow since 2024.