LEO Satellite-assisted Task Offloading for a Near Real-time Earth Observation Service

Sovit Bhandari, Thang X. Vu, and Symeon Chatzinotas
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg,
L-1855 Luxembourg. E-mail: {sovit.bhandari, thang.vu, symeon.chatzinotas}@uni.lu

***Abstract*—The next-generation regenerative payload-enabled low Earth orbit (LEO) satellites enable task offloading and delivering services to energy and computation-constrained devices in remote terrains. Recent studies on satellite-aided edge computing often focus on binary offloading scenarios, neglecting crucial system parameters such as service period, task deadline, and output size. To address these limitations, we propose a hierarchical computation framework for remote Earth observation-related services such as disaster prediction, 2D/3D scene observation, route finding, and rescue operations based on satellite images/videos. The proposed framework supports parallel and partial task offloading strategies, optimizing the communication and computation resources across the serving LEO satellite, adjacent LEO satellites, and cloud-aided gateway. Our objective is to minimize the worst-case task completion time, ensuring near real-time delivery of requested tasks. The formulated multi-time slot joint optimization problem is tackled via the proposed iterative algorithm based on successive convex approximation, demonstrating superior performance compared to baseline solutions.**

***Index Terms*—LEO satellite, task offloading, optimization, SCA, multi-time slot.**

I. Introduction

Recent advancements in Internet of Things (IoT)/user equipment (UE)-related data-intensive applications suggest that, by 2028, a minimum of 80% of the global mobile data traffic is anticipated to be attributed to video data [1]. As data-hungry applications tend to demand substantial computational efforts and energy, various research have been undertaken to partially/fully offload tasks from resource-constrained IoT/UE devices to the mobile edge computing (MEC) infrastructure of terrestrial nodes [2]. Although MEC in the terrestrial system makes it possible to push the heavy computational burden to edge computing systems, the limited computational capabilities and coverage of terrestrial edge equipment make it difficult to provide computation-intensive and sustainable services to the requesting IoT/UEs devices. Various studies have suggested alternative offloading solutions, such as utilizing unmanned aerial vehicles (UAVs) [3], [4] and high altitude platform station (HAPS) systems [5]. However, the limited coverage of both UAVs and HAPS makes them unsuitable for scenarios where operators aim to provide services to larger areas with limited resources.

Fortunately, recent advances in the fully-digital payload together with low propagation delays of low Earth orbit (LEO) satellite networks [6] provide an alternative for backhaul/fronthaul connectivity and near real-time broadband services, offering a larger field-of-view (FoV) than UAVs and HAPs, regardless of geographical terrains [7]. Several efforts have been devoted to optimize the LEO-assisted MEC systems [8]– [11]. These works, however, do not fully optimize the computation resources as they only consider the binary task offloading policy, and overlook at least one critical system parameter, such as the service period of each LEO pass or the task completion deadline. One of the reasons for the binary offloading might be due to the assumption that the tasks that are very lightweight and difficult to fragment, leading to a lack of consideration for the output size of the tasks. While the binary offloading usually works well for a single LEO satellite, it does not efficiently exploit the hierarchical architecture of the LEO satellite networks.

A. Contributions

In this paper, inspired by [7], we present a hierarchical edge computing platform for near real-time imagery processing services in remote terrains. The users request Earth observation (EO)-related services from their serving LEO satellite, which consist of processing tasks and returning tasks' output to the users, e.g., route 3D images. The requested tasks can be executed by the serving LEO satellite, or offloaded to adjacent LEO satellites and a cloud-aided gateway (GW) [12]. Our focus is on optimizing computation and communication resources, considering key factors such as service duration, task output size, and the maximum tolerable deadline. Our contributions can be summarized as follows:

- We formulate a multi-time scale optimization framework in the LEO satellite-assisted edge computing platform, which employs parallel and partial task-offloading methods to optimize the computation and communication resources. This optimization takes into account realistic system scenar-

ios, including the LEO satellite's service duration, tasks' requirements, as well as communications and energy constraints. The formulated problem belongs to the difficult class of non-linear mixed-integer non-convex optimization problems, which is generally NP-hard.

- We propose to solve the joint optimization problem via two sub-problems: (i) the first sub-problem determines the nodes where the tasks can be offloaded and the offloading decisions, utilizing binary relaxation with a penalty function and solved through successive convex approximation (SCA)-based iterative algorithm that converges numerically to the optimal solutions obtained via branch-and-bound (BnB) methods; (ii) the second sub-problem optimizes the computation capability of the access nodes and downlink delivery bandwidth. This iterative approach provides a sub-optimal solution to the original joint problem.
- Finally, the advantages of the proposed framework are demonstrated via numerical results based on a realistic-scenario synthetic dataset. Results show that our proposed method reduces the task computation time by at least 16% compared to existing reference solutions.

II. System Model

We consider a hierarchical (three-tier) LEO satellites task offloading system, comprising a serving LEO satellite (first-tier), adjacent LEO satellites including both intra- and inter-orbit satellites (second-tier), and the GW (third-tier) [7]. The FoV of the multi-orbital LEO satellites is considered to be non-overlapping, therefore the ground users are connected to only one serving LEO satellite at a time. This system is designed to dynamically transfer a portion of tasks from the serving LEO satellite to both adjacent LEO satellites and the GW on traffic load, ensuring near real-time services in remote terrains, as depicted in Fig. 1.

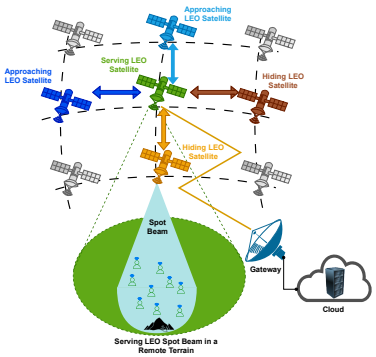

Fig. 1. LEO satellite assisted task offloading system in remote terrain.

The serving LEO satellite is assumed to establish direct connections with cloud-aided GW and M adjacent LEO satellites, two in the same orbit and the remaining $M-2$ in adjacent orbits, for task offloading. For ease of presentation, let $n \in \{0,1,2\}$ denote the nodes responsible for managing UE-requested tasks, in which $n = 0$ referring to the serving LEO satellite; $n = 1$ referring to the adjacent LEO satellites; and $n = 2$ referring to the cloud-aided GW. Each service duration T_{serve} of the serving satellite is equally divided into T time slots of equal duration Δ, and we denote $\mathcal{T} = \{1,\ldots,t,\ldots,T\}$ as the set of time slots.

Let $\mathcal{U}_t = \{1,2,\ldots,u,\ldots,U_t\}$ be the set of U_t UEs asking for the service at the time slot t within the current service duration, i.e., T_{serve}. For simplicity, we use the index i to represent elements of the set $\mathcal{U}_t$. Let $\mathcal{A}_{i,t} = \{S_{i,t}, s_{i,t}, c_{i,t}, \tau_{i,t}\}$ represent the tuples of tasks requested by the UE i at time slot t, where $S_{i,t}$ is the input size in bits, $s_{i,t}$ is the desired output size in bits, $c_{i,t}$ is the computational cycle required to execute the task in cycles/second, and $\tau_{i,t}$ is the maximum tolerance time in seconds to complete the task.

A. Communication Model

This subsection presents the transmissions between the serving LEO satellite (node $n = 0$) and the UEs, adjacent LEOs, and the GW.

1) Communication with the Requesting UEs: We focus on the transmission of the task results from the serving LEO satellite to the UEs. The transmission of the task inputs from the UEs to the serving LEO satellite is omitted for two reasons: i) this transmission does not impact the task allocation optimization, as all the computation will be executed at the satellites and/or GW; ii) the task's input size (e.g., location coordinates) is much smaller than the output size (e.g., 2D/3D images/videos). The achievable transmit rate from the serving LEO satellite to the UE i at the time slot t can be written as:

$$r_{i,t}^{(0)} = b_{i,t}^{(0)} \log_2\left(1 + \frac{p_{i,t}^{(0)} |h_{i,t}^{(0)}|^2}{b_{i,t}^{(0)} N_0}\right), \quad \forall i,t, \qquad (1)$$

where $b_{i,t}^{(0)}$ is the bandwidth allocated to UE i at time slot t, $h_{i,t}^{(0)} \in \mathbb{C}$ is the effective downlink channel coefficient from the spot beam of the serving LEO satellite to UE i, $p_{i,t}^{(0)} \triangleq \frac{P_{UE,\Sigma}^{(0)}}{U_t}$ is the transmit power of the serving LEO satellite for single-antenna UE i, wherein $P_{UE,\Sigma}^{(0)}$ stands for the total transmit power of the serving LEO satellite at time slot t. Additionally, N_0 represents the additive white Gaussian noise density. The downlink channel coefficient between the serving LEO satellite and UE i after the Doppler compensation can be modeled using a Rician fading channel as in [6].

2) Communication with the Adjacent LEO Satellites: When needed, the serving satellite offloads parts of the task to one of neighboring LEO satellites. The inter-satellite link (ISL) rate between the serving LEO satellite and offloaded adjacent LEO satellite, represented as $r_{i,t}^{(1)}$ bits-per-second (bps) for handling the request of UE i at time slot t, is assumed to be fixed [8]– [11] for simplicity[1] and is constrained by the total ISL rate R_{ISL}. This total ISL rate is defined as $R_{ISL} \triangleq \sum_{i \in U_t} r_{i,t}^{(1)}$ for all t.

3) Communication with the Cloud-aided GW : The feeder link rate $r_{i,t}^{(2)}$ bps between the serving LEO satellite and the cloud-aided GW for handling UE i request at time slot t is assumed to be static, achievable via power allocation [8]– [11], and constrained by the total feeder link rate R_{GW}. This total feeder link rate is expressed as $R_{GW} \triangleq \sum_{i \in U_t} r_{i,t}^{(2)}$ for all t.

B. Computation Model

Denote $\beta_{i,t}^{(n)} \in [0,1], n = 0,1,2$ as the fraction task offloading for user i at time slot t performed at the serving LEO, adjacent LEO, and the GW, respectively. Similarly, $f_{i,t}^{(n)}, n = 0,1,2$ denotes the computation capacity devoted for user i at time slot t.

1) Computing at the Serving LEO Satellite: The computation latency in performing a portion of the user-assigned task by the serving LEO satellite at time slot t can be written as:

$$l_{i,t}^{(0)} = \beta_{i,t}^{(0)} c_{i,t} / f_{i,t}^{(0)}, \quad \forall i,t. \tag{2}$$

The energy consumption in the serving LEO satellite while performing the requested task i at time slot t, $\forall i,t$ can be written as:

$$e_{i,t}^{(0)} = \underbrace{\sum_{i \in \mathcal{U}_t} \beta_{i,t}^{(0)} \kappa^{(0)} (f_{i,t}^{(0)})^2 c_{i,t}}_{\text{Processing}} + \underbrace{\frac{\beta_{i,t}^{(1)} p_{i,t}^{(1)} S_{i,t}}{r_{i,t}^{(1)}}}_{\text{Dataset Tx. to Node 1}} + \underbrace{\frac{\beta_{i,t}^{(2)} p_{i,t}^{(2)} S_{i,t}}{r_{i,t}^{(2)}}}_{\text{Dataset Tx. to Node 2}} + \underbrace{\frac{(\beta_{i,t}^{(0)} + \beta_{i,t}^{(1)} + \beta_{i,t}^{(2)}) p_{i,t}^{(0)} s_{i,t}}{r_{i,t}^{(0)}}}_{\text{Result Delivery to UEs}}, \tag{3}$$

where $\kappa^{(0)}$ is the energy factor of the serving LEO (node 0), $r_{i,t}^{(n)}$ is defined in Section II-A, and $p_{i,t}^{(0)}, p_{i,t}^{(1)}, p_{i,t}^{(2)}$ are the power used for communications to the UE, ISL, and to the GW, respectively.

2) Computing at the Adjacent LEO Satellite: The computational time delay in performing a portion of the user-assigned task by the neighboring LEO satellite at time slot t can be written as:

$$l_{i,t}^{(1)} = \underbrace{\beta_{i,t}^{(1)} \frac{c_{i,t}}{f_{i,t}^{(1)}}}_{\text{Processing}} + \underbrace{\alpha_{i,t}^{(1)} \frac{2 d_{0,t}^{(1)}}{c_0}}_{\text{Propagation}} + \underbrace{\frac{\beta_{i,t}^{(1)} (S_{i,t} + s_{i,t})}{r_{i,t}^{(1)}}}_{\text{Dataset Tx.}}, \forall i,t, \tag{4}$$

where $d_{0,t}^{(1)}$ represents the ISL distance, $\alpha_{i,t}^{(1)} \in \{0,1\}$ is a binary variable indicating whether task i has been offloaded to the neighboring LEO ($\alpha_{i,t}^{(1)} = 1$) or not ($\alpha_{i,t}^{(1)} = 0$), and c_0 is the speed of light.

The energy consumption in the adjacent LEO satellite while performing the requested task i at the time slot t can be written as:

$$e_{i,t}^{(1)} = \underbrace{\sum_{i \in \mathcal{U}_t} \beta_{i,t}^{(1)} \kappa^{(1)} (f_{i,t}^{(1)})^2 c_{i,t}}_{\text{Processing}} + \underbrace{\frac{\beta_{i,t}^{(1)} p_{0,i,t}^{(1)} s_{i,t}}{r_{i,t}^{(1)}}}_{\text{Result Tx. to Node 0}}, \forall i,t, \tag{5}$$

where $\kappa^{(1)}$ represents the energy factor of node 1, and $p_{0,i,t}^{(1)}$ is the power used by node 1 to communicate with node 0, transferring a part of the completed task i from node 1 to node 0.

3) Computing at the GW: The computation time delay in performing a portion of the user-assigned task by the cloud-aided GW at time slot t can be written as:

$$l_{i,t}^{(2)} = \underbrace{\beta_{i,t}^{(2)} \frac{c_{i,t}}{f_{i,t}^{(2)}}}_{\text{Processing}} + \underbrace{\alpha_{i,t}^{(2)} \frac{2 d_{0,t}^{(2)}}{c_0}}_{\text{Propagation}} + \underbrace{\frac{\beta_{i,t}^{(2)} (S_{i,t} + s_{i,t})}{r_{i,t}^{(2)}}}_{\text{Dataset Tx.}}, \forall i,t, \tag{6}$$

where $d_{0,t}^{(2)}$ represents the slant distance between the serving LEO and GW, and $\alpha_{i,t}^{(2)} \in \{0,1\}$ is a binary variable indicating whether task i has been offloaded to the GW ($\alpha_{i,t}^{(2)} = 1$) or not ($\alpha_{i,t}^{(2)} = 0$).

The energy consumption at the GW is omitted since it usually supplied by constant power source.

C. Return Link Latency

The return-link latency in transmitting the desired result after the completion of the task i to the requesting UE from the serving LEO satellite at the time slot t can be written as:

$$l_{i,t}^{rtn} = \underbrace{\frac{(\sum_{n \in \mathcal{N}} \beta_{i,t}^{(n)}) s_{i,t}}{r_{i,t}^{(0)}}}_{\text{Result Tx.}} + \underbrace{\frac{d_{i,t}^{(0)}}{c}}_{\text{Propagation}}, \forall i,t, \tag{7}$$

where $r_{i,t}^{(0)}$ is given in (1), and $d_{i,t}^{(0)}$ is the slant distance between node 0 and UE i at time slot t.

III. Problem Formulation and Proposed Solution

We aim to optimize computational resources and downlink delivery bandwidth to fulfill all requested tasks in the minimum possible time, providing flexibility across multiple time slots. This is constrained by the service duration, task deadline, downlink delivery quality of service (QoS) requirement, and the total energy and computational resources of the participant

[1]This static rate assumption can be easily achieved by power adaption to the relative distance dynamics of the ISL.

access nodes. Mathematically, the multi-objective optimization problem can be formulated as follows:

$$\mathcal{P}: \max_{\{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{f},\boldsymbol{b}\}} \; w_1 \sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} \sum_{n\in\mathcal{N}} \beta_{i,t}^{(n)} - (1-w_1)\times L(\boldsymbol{\Pi}) \quad (8a)$$

$$\text{s.t. } l_{i,t}^{proc} + l_{i,t}^{rtn} \le \Gamma_{i,t}, \quad \forall n,i,t, \quad (8b)$$

$$r_{i,t}^{(0)} \ge \eta_{req}^{(0)}, \forall i,t; \quad \sum_{i\in\mathcal{U}_t} b_{i,t}^{(0)} \le B_{UE,\Sigma}^{(0)}, \forall t, \quad (8c)$$

$$\sum_{t\in\mathcal{T}'} \sum_{n\in\mathcal{N}} \beta_{i,t}^{(n)} \le 1, \quad \forall i, \quad (8d)$$

$$\beta_{i,t}^{(n)} \le \alpha_{i,t}^{(n)}, \quad \forall n\in\mathcal{N}\backslash\{0\}, i,t, \quad (8e)$$

$$\sum_{i\in\mathcal{U}_t} f_{i,t}^{(n)} \le F_{\Sigma}^{(n)}, \quad \forall n,t, \quad (8f)$$

$$f_{i,t}^{(n)} \le \beta_{i,t}^{(n)} F_{\Sigma}^{(n)}, \quad \forall n,i,t, \quad (8g)$$

$$\sum_{i\in\mathcal{U}_t} e_{i,t}^{(n)} \le E_{\Sigma}^{(n)}, \quad \forall n\in\mathcal{N}\backslash\{2\}, t, \quad (8h)$$

$$0 \le \beta_{i,t}^{(n)} \le 1, \quad \forall n,i,t, \quad (8i)$$

$$\alpha_{i,t}^{(n)} \in \{0,1\}, \quad \forall n\in\mathcal{N}\backslash\{0\}, i,t, \quad (8j)$$

where $\boldsymbol{\alpha} \triangleq \{\alpha_{i,t}^{(n)}\}_{\forall n\in\{1,2\},i,t}$, $\boldsymbol{\beta} \triangleq \{\beta_{i,t}^{(n)}\}_{\forall n,i,t}$, $\boldsymbol{f} \triangleq \{f_{i,t}^{(n)}\}_{\forall n,i,t}$, and $\boldsymbol{b} \triangleq \{b_{i,t}^{(0)}\}_{\forall i,t}$ are the short-hand notations; $\boldsymbol{\Pi} \triangleq (\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{f},\boldsymbol{b})$; $\Gamma_{i,t} \triangleq \max\{\min\{\Delta, \tau_{i,t}-(t-1)\Delta\},0\}$; $l_{i,t}^{proc} \triangleq \max\{l_{i,t}^{(0)},\ldots,l_{i,t}^{(n)},\ldots,l_{i,t}^{(N)}\}$; $\mathcal{T}'$ represents the list of time slots required to complete tasks within the overall time frame $\mathcal{T}$, i.e., $\mathcal{T}' \subseteq \mathcal{T}$; w_1 is the weight factor to control the multi-objective optimization problem; $\eta_{req}^{(0)}$ is the minimum QoS requirement from the UEs; $B_{UE,\Sigma}^{(0)}$ is the total downlink bandwidth allocated for the UEs from the serving satellite; $F_{\Sigma}^{(n)}$ is the total computational capabilities of node n; and $E_{\Sigma}^{(n)}$ represents the total energy of each node n excluding node 2 (the GW).

The second part of the multi-objective function, i.e., $L(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{f},\boldsymbol{b})$ in problem (8), is defined as the overall worst-case latency for delivering the user's requested task in near real-time and can be expressed as follows:

$$L(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{f},\boldsymbol{b}) = \max_i \textstyle\sum_{t\in\mathcal{T}'} \left\{l_{i,t}^{proc} + l_{i,t}^{rtn}\right\}, \quad (9)$$

where $l_{i,t}^{proc}$ represents the worst-case parallel task processing time and $l_{i,t}^{rtn}$ denotes the total latency accumulated due to transmission and propagation delays (Section II-B and II-C).

In problem $\mathcal{P}$, constraint (8b) ensures completion of the UE-requested task within minimum time slots and duration, meeting UE's tolerance limit and node 0's service period. Constraint (8c) guarantees minimum QoS requirement and restricts the sum of sub-channel downlink bandwidth for UE. Constraint (8d) ensures the total task performed does not exceed the requested amount. Constraint (8e) ensures the task portion is only offloaded to selected nodes. Constraints (8f) and (8g) regulate the computational capabilities assigned for performing each UE-requested task in nodes 0–2. Constraint (8h) limits energy consumption in nodes 0 and 1 during task execution. Constraint (8i) bounds partial task offloading variable within [0, 1], while constraint (8j) controls the binary nature of node selection variables for offloading.

Difficulty to solve problem $\mathcal{P}$: The problem $\mathcal{P}$ is challenging to solve due to the non-convexity of its objective function and the constraints (8b),(8h), and (8j), respectively. Moreover, it is non-convex mixed-integer nonlinear programming (MINLP) and is NP-hard, primarily due to the binary nature of the offloading node selection variables $\{\alpha_{i,t}^{(n)}\}$.

A. Proposed Solution

To achieve an efficient solution for problem $\mathcal{P}$, we decompose the original optimization problem into two sub-problems:

1) Solving for optimal values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$:

$$\mathcal{P}_1: \max_{\{\boldsymbol{\alpha},\boldsymbol{\beta}\}} w_1 \sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} \sum_{n\in\mathcal{N}} \beta_{i,t}^{(n)} - (1-w_1)\times L(\boldsymbol{\alpha},\boldsymbol{\beta}) \quad (10a)$$

$$\text{s.t. (8b), (8d), (8e), (8g) – (8j),}$$

which is a MINLP problem. We are interested in converting this problem to the mixed integer linear programming (MILP) problem. The non-linearity in the constraint (8b) can be converted into the linear form by consideration of the slack variable $\{x_{i,t}\}$ and reformulate constraint (8b) as:

$$l_{i,t}^{(n)} \le x_{i,t}, \quad \forall n,i,t; \; x_{i,t} + l_{i,t}^{rtn} \le \Gamma_{i,t}, \quad \forall i,t. \quad (11)$$

With the inclusion of the slack variable $\{x_{i,t}\}$, the non-linear component of the objective function (10a), i.e., $\max_i \left\{\sum_{t\in\mathcal{T}'} l_{i,t}^{proc} + l_{i,t}^{rtn}\right\}$, denoted as $L(\boldsymbol{\alpha},\boldsymbol{\beta})$, can be expressed as $\max_i \left\{\sum_{t\in\mathcal{T}'} x_{i,t} + l_{i,t}^{rtn}\right\}$. However, this expression remains non-linear due to the $\max_i$ term. To linearize it, a slack variable y is introduced, which is taken as the upper bound. Consequently, the problem $\mathcal{P}_1$ can be equivalently reformulated as:

$$\mathcal{P}_1^{'}: \min_{\{\boldsymbol{\alpha},\boldsymbol{\beta}\}} \; w_1 \sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} \sum_{n\in\mathcal{N}} \beta_{i,t}^{(n)} - (1-w_1)y \quad (12a)$$

$$\text{s.t. } \max_i \left\{\sum_{t\in\mathcal{T}'} x_{i,t} + l_{i,t}^{rtn}\right\} \le y, \quad \forall,i, \quad (12b)$$

$$(11), (8d), (8e), (8g) – (8j). \quad (12c)$$

Since the objective function, as well as all the constraints, are linear and involve fractional/binary variables in (12), it can be solved using the standard BnB method. To address this computational challenge, we choose to relax the binary variables, converting them into continuous ones. This can be achieved by adding a penalty function $\Psi^{(n)} = \left(\sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} (\alpha_{i,t}^{(n)})^2\right) - \left(\sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} \alpha_{i,t}^{(n)}\right)$ to the original objective function, i.e., (12a), to facilitate convergence. It's crucial to note that $\Psi^{(n)}$ takes a quadratic form, and we linearize this quadratic penalty function by applying a first-order Taylor expansion around a reference point $\bar{\alpha}_{i,t}^{(n)}$ as:

$$\Psi^{(n)} \triangleq \left(\sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} 2\alpha_{i,t}^{(n)}\right)\left(\sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} \bar{\alpha}_{i,t}^{(n)}\right) - \left(\sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} (\bar{\alpha}_{i,t}^{(n)})^2\right) - \left(\sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} \alpha_{i,t}^{(n)}\right), \; \forall n. \quad (13)$$

By using the approximated penalty function (13) in (12a), $\mathcal{P}_1^{'}$ with a penalty parameter w_2 can be formulated as:

$$\mathcal{P}_1^{'}: \min_{\{\boldsymbol{\alpha},\boldsymbol{\beta}\}} w_1 \sum_{t\in\mathcal{T}'} \sum_{i\in\mathcal{U}_t} \sum_{n\in\mathcal{N}} \beta_{i,t}^{(n)} - (1-w_1)y + w_2 \sum_{n\in\{1,2\}} \Psi^{(n)} \quad (14a)$$

$$\text{s.t. } 0 \le \alpha_{i,t}^{(n)} \le 1 \quad \forall n,i,t; \; (12b), (12c).$$

It is observed that the problem (14), which is the relaxed version of (10), is an linear programming problem because of the linear objective function and linear constraints. Thus, problem (14) can be solved efficiently using standard methods, e.g., the interior point (IP) [11]. We note that problem

(14) provides a sub-optimal solution to problem (10) since it satisfies all constraints of this problem and also depends largely on the initialization of the parameter $\bar{\boldsymbol{\alpha}}$. Therefore, we propose Algorithm 1 to solve (10), where obj$(\boldsymbol{\alpha}_{i_1}, \boldsymbol{\beta}_{i_1})$ and obj$(\boldsymbol{\alpha}_{i_1-1}, \boldsymbol{\beta}_{i_1-1})$ denotes the objective function (14a) at iterations i_1 and $i_1 - 1$, respectively.

2) Solving for optimal values of $\boldsymbol{f}$ and $\boldsymbol{b}$:

$$\mathcal{P}_2 : \min_{\{\boldsymbol{f},\boldsymbol{b}\}} \max_{i} \left\{ \sum_{t \epsilon \mathcal{T}'} l_{i,t}^{proc} + l_{i,t}^{rtn} \right\} \tag{15a}$$

$$\text{s.t. (8b), (8c), (8f) – (8h).}$$

It is observed that the problem $\mathcal{P}_2$ is a non-linear convex problem, which can be solved efficiently using the IP method. The IP method yields values for $\boldsymbol{f}^*$ and $\boldsymbol{b}^*$. These values are then utilized as input for initializing the parameters $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{b}}$ to solve the optimization problem (10) using Algorithm 1.

Algorithm 1 Iterative Algorithm to Solve (10)

Input: U_t, w_1, $\boldsymbol{f}$, $\boldsymbol{b}$, $\tau_{i,t}$, Δ, $B_{UE,\Sigma}^{(0)}$, $F_{\Sigma}^{(n)}$, $E_{\Sigma}^{(n)}$, $S_{i,t}$, $s_{i,t}$, $c_{i,t}$, c_0, $d_{i,t}^{(0)}$, $d_{0,t}^{(1)}$, $d_{0,t}^{(2)}$, $p_{i,t}^{(n)}$, $p_{0,i,t}^{(1)}$, $p_{0,i,t}^{(2)}$, $\kappa^{(n)}$, $r_{i,t}^{(1)}$, $r_{i,t}^{(2)}$, N_0

Output: $\alpha_{i,t}^{(1)*}$, $\alpha_{i,t}^{(2)*}$, $\beta_{i,t}^{(0)*}$, $\beta_{i,t}^{(1)*}$, $\beta_{i,t}^{(2)*}$

Initialization: $\bar{\alpha}_{i,t}^{(1)}$, $\bar{\alpha}_{i,t}^{(2)}$, $i_1 = 1$, $I_{max} = 10$, $\epsilon = 10^{-2}$, $e_1 = 100$

1: **while** $e_1 > \epsilon$ and $i_1 < I_{max}$ **do**
2: // Solve (14) to get $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}^*$
3: Cmp. obj$(\boldsymbol{\alpha}_{i_1}, \boldsymbol{\beta}_{i_1})$
4: Cmp. $e_1 = |\text{obj}(\boldsymbol{\alpha}_{i_1}, \boldsymbol{\beta}_{i_1}) - \text{obj}(\boldsymbol{\alpha}_{i_1-1}, \boldsymbol{\beta}_{i_1-1})|$
5: Update obj$(\boldsymbol{\alpha}_{i_1-1}, \boldsymbol{\beta}_{i_1-1}) \leftarrow$obj$(\boldsymbol{\alpha}_{i_1}, \boldsymbol{\beta}_{i_1})$; $\bar{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}^*$; $i_1 \leftarrow i_1 + 1$
6: **end while**

IV. Performance Evaluation on Realistic System Parameters

A. LEO Satellite Footprint

The Starlink LEO satellite 4280 is assumed to be in orbit just above Mt. Everest, Nepal at time t_0 [14]. It is at an altitude (H_s) of 545 km just above Earth's surface and is considered to be $d_{0,t}^{(1)}$ distance apart from the $M = 4$ adjacent LEO satellites, i.e., 2 in the same orbit and the remaining 2 in the adjacent orbits. The FoV of one of the spot beams of the serving LEO satellite is shown in Fig. 2.

B. Gen. of the Synthetic Dataset Based on Realistic Info.

Task requests are synthetically generated based on realistic information due to the absence of suitable open-source datasets in this domain. According to [15], during a specific window period, at least 500 tourists daily journey to Everest Base Camp (EBC), with 208 individuals aiming to climb Mt. Everest. The primary locations during the ascent include EBC, Camp-I, Camp-II, Camp-III, Camp-IV, and Mt. Everest. Requests to the LEO satellite are anticipated from users near these locations. At any given time, there are 292 users at EBC, and 208 users are evenly distributed in Camp-I, Camp-II, Camp-III, Camp-IV, and Mt. Everest. Considering 2% of active users every minute at both EBC and ascending Mt. Everest, the task arrival rate vector is denoted as $\boldsymbol{\lambda} = [5.8, 0.8, 0.8, 0.8, 0.8, 1]\text{min}^{-1}$.

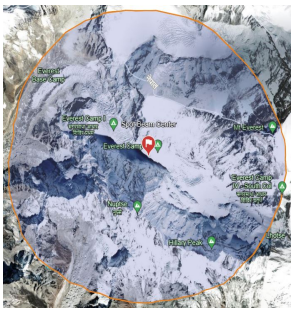

Fig. 2. FoV of the serving LEO satellite's spot beam.

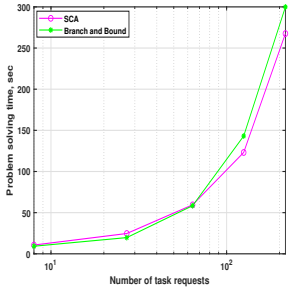

Fig. 3. Time-complexity of the proposed algorithm.

TABLE I
Simulation Parameters

Parameters	Value
$\tau_{i,t}$, Δ	60 sec, 60 sec
$S_{i,t}$, $s_{i,t}$	U [10 MB, 20 MB] [16], $0.6 \times S_{i,t}$ [17]
$c_{i,t}$	$\{0.05, 0.075\}$ Kcycles/bit [18]
$F_{\Sigma}^{(0)}$, $F_{\Sigma}^{(1)}$, $F_{\Sigma}^{(2)}$	3 GHz [10], 3 × 4 GHz, 12 GHz
$E_{\Sigma}^{(0)}$, $E_{\Sigma}^{(1)}$, $\kappa^{(n)}$	5000 MJ [10], 4 × 5000 MJ, 10^{-28} [10]
R_{ISL}, R_{GW}	10 Gbps [10], 300 Mbps [10]
$d_{0,t}^{(1)}$, $d_{0,t}^{(2)}$	1000 km [10], 3000 km
$P_{ISL}^{(0)}$, $P_{GW}^{(0)}$, $P_{UE}^{(0)}$	30 dBW [10], 20 dBW [10], 25 dBW [6]
$B_{UE,\Sigma}^{(0)}$, f_c	100 MHz [6], 2 GHz [6]
$\eta_{req}^{(0)}$, N_0	50 Mbps [6], -203.9772 dBW/Hz

C. Performance Evaluation

We present numerical results considering a scenario where an LEO satellite employs only one steerable adaptive spot beam [6]. The positions of UEs making task requests are distributed based on $\boldsymbol{\lambda}$ across six major locations within the coverage area, as shown in Fig. 2. The simulation parameters are summarized in Table I. The simulation results are averaged over 200 random channel realizations. We compare the proposed framework with the following references:

- *Equal Task Portion Offloading:* This is a proposed task offloading scenario that involves an equal distribution of resources without optimizing them.
- *Binary Offloading:* This is a binary offloading method optimizing computational resources and downlink delivery bandwidth [8]– [11].
- *Fixed Computation and Communication:* This is a fractional task offloading scenario without optimizing computational resources and downlink bandwidth [19].
- *Without Offloading:* No task offloading occurs; all the computations are done on the serving LEO satellite.

To demonstrate the time complexity of our proposed algorithms, Fig. 3 shows that both the SCA and BnB methods exhibit exponential complexity as the problem size increases. However, for larger problem sizes, the SCA method demonstrates comparatively faster convergence compared to the BnB method while providing the sub-optimal solution.

Fig. 4 depicts the trend of the proposed method based on SCA for selected time slots, where the number of task requests at each time slot varies according to the Poisson distribution. It also showcases the superiority of the proposed method over the reference methods in terms of task completion time, with

both trends influenced by the number of task request arrivals at each time slot.

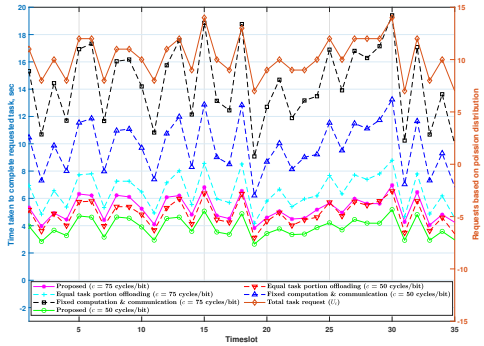

Fig. 4. Trend of task completion time for different time slots.

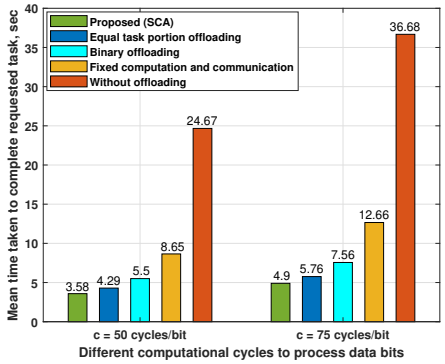

Fig. 5. Mean task completion time for different computational cycles.

Fig. 5 portrays the mean time required to fulfill the requested tasks using the proposed method and compares it with other reference methods, taking into account diverse computational requirements for processing data bits. With a computational requirement of 50 cycles per bit, the proposed method achieves a mean completion time of 3.58 seconds, surpassing equal task portion offloading, binary offloading, fixed computation and communication, and without offloading by 16.55%, 34.90%, 58.61%, and 85.48%, respectively. As the computational requirement increases to 75 cycles per bit, the mean completion time for the proposed method increases by 36.87%, while equal task portion offloading, binary offloading, fixed computation and communication, and without offloading increase by 34.27%, 37.45%, 46.36%, and 48.68%, respectively. This indicates a sub-linear increase in the time taken to complete the requested task for both the proposed method and reference methods with the increment in computational requirements for processing data bits.

V. Conclusion

In this paper, we have developed a satellite-assisted Earth observation framework capable of partial and parallel task offloading. It dynamically optimizes communication and computation resources of access nodes, including serving LEO satellites, adjacent LEO satellites, and a cloud-aided GW while addressing requested task load from users' remote terrains. Simulations conducted with realistic parameters show that our proposed approach outperforms other reference offloading approaches by at least 16%. Additionally, the superior performance of the SCA-based approach over the BnB method is highlighted, particularly for a large number of variables, in terms of computational efficiency.

Acknowledgment

This work is supported by the Luxembourg National Research Fund (FNR) via project INSTRUCT, ref. IPBG19/14016225/INSTRUCT, and project RUTINE, ref. C22/IS/17220888/RUTINE.

References

[1] "Traffic by application – mobility report," Ericsson. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/traffic-by-application.

[2] A. Islam et al., "A Survey on Task Offloading in Multi-access Edge Computing,"*Jrnl. of Sys. Arch.*, vol. 118, 2021.

[3] A. Mahmood et al., "Optimizing Computational and Communication Resources for MEC Network Empowered UAV-RIS Communication," *2022 IEEE Globecom Workshops*, Rio de Janeiro, Brazil, 2022, pp. 974–979.

[4] X. Dai et al., "UAV-Assisted Task Offloading in Vehicular Edge Computing Networks,"*IEEE Trans. on Mobile Computing*, vol. 23, no. 4, pp. 2520–2534.

[5] Q. Ren et al., "Caching and Computation Offloading in High Altitude Platform Station (HAPS) Assisted Intelligent Transportation Systems," in *IEEE Trans. on Wirel. Commun.*, vol. 21, no. 11, pp. 9010–9024, 2022.

[6] S. Bhandari et. al, "User-centric Flexible Resource Management Framework for LEO satellites with fully regenerative payload," in *IEEE Jrnl. on Selected Areas in Commun.*, vol. 42, no. 5, pp. 1246–1261.

[7] I. Leyva-Mayorga et al., "Satellite Edge Computing for Real-Time and Very-High Resolution Earth Observation," in *IEEE Trans. Commun.*, vol. 71, no. 10, pp. 6180–6194.

[8] M. Tong et al., "Inter-satellite cooperative offloading decision and resource allocation in Mobile Edge Computing-enabled satellite–terrestrial networks," *Sensors*, vol. 23, no. 2, p. 668, 2023.

[9] Y. Hao et al., "Joint Communication, Computing, and Caching Resource Allocation in LEO Satellite MEC Networks," in *IEEE Access*, vol. 11, pp. 6708–6716, 2023.

[10] H. Zhang et al., "Satellite Edge Computing With Collaborative Computation Offloading: An Intelligent Deep Deterministic Policy Gradient Approach," in *IEEE Internet of Things Jrnl.*, vol. 10, no. 10, pp. 9092–9107, 2023.

[11] C. Mei et al., "Joint Task Offloading and Resource Allocation for space–Air–Ground Collaborative Network," in *Drones*, vol. 7, no. 7, p. 482, 2023.

[12] "Satellite-as-a-service: The future of satellite network communications: Eutelsat," eutelsat. [Online].

[13] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge Univ. Press, 2004.

[14] "Technical details for satellite starlink-4280," N2YO.com. [Online].

[15] "How many people have climbed Mount Everest?," Ian Taylor Trekking. [Online].

[16] Y. Jiang et al., "A collaborative optimization strategy for computing offloading and resource allocation based on multi-agent deep reinforcement learning," *Computers and Electrical Engineering*, vol. 103, p. 108278, 2022.

[17] R. C. Gonzalez and R. E. Woods, Digital Image Processing. Upper Saddle River: Pearson, 2019.

[18] T. Deng et al., "Task offloading based on edge collaboration in MEC-enabled IoV networks," in *Jrnl. of Communications and Netwks.*, vol. 25, no. 2, pp. 197–207.

[19] Q. Tang et al., "Computation Offloading in LEO Satellite Networks With Hybrid Cloud and Edge Computing," in *IEEE IoT Jrnl.*, vol. 8, no. 11, pp. 9164–9176, 2021.