GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10001422

Joint Computation Offloading, UAV Trajectory, User Scheduling, and Resource Allocation in SAGIN

Minh Dat Nguyen, Long Bao Le, and André Girard

***Abstract*—In this paper, we study the computation offloading problem in space-air-ground integrated networks (SAGIN), where joint optimization of partial computation offloading, unmanned aerial vehicles (UAVs) trajectory control, user scheduling, computation and resource allocation is performed. Specifically, the considered SAGIN employs multiple UAV-mounted edge servers with controllable UAV trajectory and a cloud sever which can be reached by ground users (GUs) via multi-hop low-earth-orbit (LEO) satellite communications. This design aims to minimize the weighted energy consumption of the GUs and UAVs while satisfying the maximum delay constraints of underlying computation tasks. To tackle the underlying non-convex mixed integer non-linear optimization problem, we use the alternating optimization approach where we iteratively solve five sub-problems, namely user scheduling, partial offloading control and bit allocation over time slots, computation resource, bandwidth allocation, and multi-UAV trajectory control until convergence. In addition, the successive convex approximation (SCA) method is employed to convexify and solve the non-convex bandwidth allocation and UAV trajectory control sub-problems. Via extensive numerical studies, we illustrate the effectiveness of our proposed design compared to baselines under different network settings.**

***Index Terms*—Multi-hop satellite communication, UAV trajectory, mobile edge computing, user scheduling, resource allocation**

I. Introduction

Future wireless networks are expected to provide higher capacity and much lower latency, better communications reliability, and to support emerging Internet of Things (IoTs) applications. To this end, low-earth-orbit (LEO) satellites, unmanned aerial vehicle (UAV) communications, and mobile edge computing have been considered as promising technologies [1]. In particular, LEO satellite communications with many advantages such as low propagation delay, high communication rate, and seamless communication services to wide geographical areas where ground base stations are not available or damaged by natural disasters have attracted great attention [2]. In addition, space-air-ground integrated networks (SAGIN) have emerged as an effective means to provide high quality and ubiquitous communications by leveraging the complementarity and strength of the space, air, and ground network segments.

Recent studies on the uplink communications and computation offloading in the SAGIN have been investigated [3]–[5]. In particular, uplink communications with ultra-dense LEO satellite constellations and multiple UAVs were studied in [3] to optimize the data gathering efficiency. Meanwhile, joint optimization of task scheduling and computation resource allocation was considered in [4] where multiple satellites and fixed UAVs' trajectories were assumed with the objective of minimizing the total system cost in terms of the task delay, users' energy consumption, and server usage costs. Moreover, the SAGIN setting with one satellite and multiple UAVs was considered in [5] where the design objective is to minimize the maximum delay experienced by the ground users (GUs) by jointly optimizing UAV-device association, task assignment, power control, bandwidth allocation, computation resource, and UAV placement.

To the best of our knowledge, none of the previous work has studied the computation offloading for the SAGIN-based edge-cloud computing system considering user scheduling during the UAV flight period to satisfy the maximum delay requirements of underlying computation tasks. Moreover, the multi-hop satellite communications to access the cloud server located far away from the considered terrestrial network area have been mostly ignored in the existing SAGIN-based computation offloading designs. To fill these research gaps, we study partial computation offloading in SAGIN where fractions of computation tasks from GUs are processed locally and/or offloaded and processed at the UAV-mounted edger servers and cloud server leveraging multi-hop LEO satellite communications. Our design aims to minimize the weighted energy consumption of the GUs and UAVs while satisfying the maximum delay constraints of underlying computation tasks by jointly optimizing the user scheduling, partial offloading control and bit allocation over time slots, computation resource allocation, bandwidth allocation, and UAV trajectory control. The alternating optimization approach is employed to solve the underlying non-convex mixed-integer non-linear optimization problem (MINLP). Moreover, the successive convex approximation (SCA) method is employed solve the bandwidth allocation and UAV trajectory control sub-problems. Numerical results are presented to show the impacts of different parameters including the hop counts in the multi-hop satellite communications and computation task size and the performance gains achieved by optimizing the UAV trajectory control, user scheduling, and computation offloading.

II. System Model

We study the computation offloading design in the generic SAGIN as illustrated in Fig. 1 where the terrestrial network layer provides services for K GUs, the aerial network layer employs M UAVs, and the space network layer relies on LEO satellites for connections to a distant cloud server. We denote the sets of satellites, UAVs, and GUs as $\mathcal{S} = \{1, ..., S\}$, $\mathcal{M} = \{1, ..., M\}$, and $\mathcal{K} = \{1, ..., K\}$, respectively. Partial computation offloading is considered where each GU partitions its unique computation task into three sub-tasks where the first

M. D. Nguyen, L. B. Le, and A. Girard are with INRS-EMT, University of Québec, Montréal, QC H5A 1K6, Canada (email: minh.dat.nguyen@inrs.ca, long.le@inrs.ca, andre@inrs.ca).

sub-task is processed locally and the two remaining sub-tasks are offloaded and processed at the UAV-mounted edge server and the cloud server, respectively. Moreover, the data related to the second sub-task must be transmitted from the associated GU to the connected UAV and the data related to the third sub-task must be transmitted from the GU over the multi-hop satellite communication path to the cloud server.

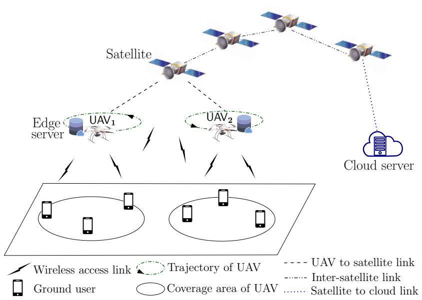

Fig. 1: SAGIN with multi-hop satellite communications.

All GUs are assumed to be located on the ground at zero altitude with fixed horizontal coordinates of $\mathbf{r}_k^{\mathsf{u}}{=}(x_k^{\mathsf{u}}, y_k^{\mathsf{u}}), \forall k{\in}\mathcal{K}$. Besides, we assume that the UAVs fly at a fixed altitude H over a flight period of $T{>}0$ seconds and we divide the flight period into N time slots, whose set is denoted as $\mathcal{N}{=}\{1,...,N\}$. Moreover, uplink transmissions from multiple GUs to their associated UAVs are enabled by using the frequency division multiple access (FDMA). Specifically, let W denote the total bandwidth available to support uplink communications from GUs to UAVs, then we assume that the available bandwidth is partitioned into orthogonal sub-bands each of which is allocated to one corresponding UAV to serve its associated GU. We denote the bandwidth allocated for UAV m as W_m^{u} then we have $\sum_{m\in\mathcal{M}} W_m^{\mathsf{u}}{=}W$. Furthermore, the computation results produce much smaller data sets than the offloading data so that we can neglect the download time of the computation results in the offloading process.

A. Computation Task Model

We assume that each GU k has a latency-constrained computation task with the corresponding parameters denoted as $U_k{=}(f_k, s_k, c_k, T_k^{\mathsf{max}})$, where f_k denotes the computation load expressed by the number of central process unit (CPU) cycles per second (CPU cycles/second), s_k (bits) represents the size of input raw data, c_k (CPU cycles/bit) denotes the computation load corresponding to 1-bit input data, and T_k^{max} (seconds) represents the maximum permissible latency of the task U_k.

We assume that the GU's computation task can be partitioned into three sub-tasks that are processed in parallel at the GU, the UAV-mounted edge server, and the cloud server reached via the multi-hop LEO satellite communication path as previously considered in [5], [6]. Then, the task processing time for GU k can be written as $T_k = \max\left\{T_k^{\mathsf{lo}}, T_k^{\mathsf{ed}}, T_k^{\mathsf{cl}}\right\}$, where $T_k^{\mathsf{lo}}, T_k^{\mathsf{ed}}$, and T_k^{cl} represent the task execution time at the GU, UAV-mounted edge server, and cloud server, respectively. This processing time must satisfy the delay constraint $T_k \leq T_k^{\mathsf{max}}$. Moreover, to model the task partitioning, we introduce variables λ_k^{lo} and λ_k^{ed}, $(0{\leq}\lambda_k^{\mathsf{lo}}, \lambda_k^{\mathsf{ed}}{\leq}1)$ that represent the fractions of input data to be processed locally at GU k and to be offloaded and processed at the UAV-mounted edge server, respectively. Hence, $(1{-}\lambda_k^{\mathsf{lo}}{-}\lambda_k^{\mathsf{ed}})$ represents the fraction of input data from GU k to be offloaded and processed at the distant cloud server.

B. UAV Trajectory Control

The UAV trajectory over the flight period T is controlled by optimizing the UAV's locations in individual time slots. Specifically, the horizontal coordinates of UAV m in time slot n is denoted as $\mathbf{q}_m[n]{=}(x_m^{\mathsf{d}}[n], y_m^{\mathsf{d}}[n])$. We assume that each UAV must come back to its initial position at the end of the flight period, i.e., $\mathbf{q}_m[1]{=}\mathbf{q}_m[N], \forall m{\in}\mathcal{M}$. In addition, the slot interval $\Delta t{=}\frac{T}{N}$ is chosen to be sufficiently small so that the UAVs' locations are within a bounded small neighborhood in each time slot even at the maximum flight speed $V_{\max}$ expressed in meter/second (m/s). Hence, the UAVs' trajectories must meet the following constraints: $\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\|^2 \leq S_{\max}^2, n{=}1,...,N{-}1$, and $\|\mathbf{q}_m[n] - \mathbf{q}_j[n]\|^2 \geq d_{\min}^2, \forall n, m, j \neq m$, where $\|.\|$ denotes the Euclidean norm, $S_{\max} \triangleq V_{\max}\Delta t$ is the maximum horizontal distance that the UAV can travel in each time slot, $d_{\min}$ denotes the minimum allowable inter-UAV distance between UAVs to ensure collision avoidance.

C. User Scheduling

We introduce decision variables $\phi_{k,m}^{\mathsf{u}}[n]$ to capture the association between the GUs and UAVs over flight period T, where $\phi_{k,m}^{\mathsf{u}}[n] = 1$, if GU k is served by UAV m in time slot n and $\phi_{k,m}^{\mathsf{u}}[n] = 0$, otherwise. We assume that each GU can offload its computation sub-task to at most one UAV in each time slot, i.e., $\sum_{m\in\mathcal{M}} \phi_{k,m}^{\mathsf{u}}[n]{\leq}1$. Moreover, we assume each GU k is initially associated with the UAV providing the highest average received signal strength (RSS). To meet the delay requirement of the computation task of each GU, the number of *consecutive time slots* required to completely process the task of GU k can be denoted as $N_k = \lceil T_k^{\mathsf{max}}/\Delta t \rceil$, where $\lceil . \rceil$ denotes the round-up operation. We now introduce binary user scheduling variables as $\theta_k[n]$, where $\theta_k[n] = 1$, if GU k is scheduled to transmit to its associated UAV in time slot n and $\theta_k[n] = 0$, otherwise. We need to impose the following constraints on the user scheduling decisions $\sum_m \sum_{t=0}^{N_k-1} \theta_k[n{+}t]\phi_{k,m}^{\mathsf{u}}[n{+}t]{=}N_k, \forall k, n{\in}\{1,...,N - N_k\}$.

D. Computing Models

The task execution time and energy consumption are discussed in the following

1) Local Computing Model: The local task execution time at GU k can be expressed as

$$T_k^{\mathsf{lo}} = \frac{\lambda_k^{\mathsf{lo}} s_k c_k}{f_k}. \tag{1}$$

This processing delay must satisfy $T_k^{\mathsf{lo}} \leq T_k^{\mathsf{max}}$. The energy consumption due to local task execution can be calculated as

$$E_k^{\mathsf{lo}} = \kappa \lambda_k^{\mathsf{lo}} s_k c_k (f_k)^2, \tag{2}$$

where κ denotes the effective switched capacitance depending on the chip architecture [7].

2) UAV-Mounted Edge Server Model: For the partitioned sub-tasks offloaded to the UAVs, let $l_k^{\mathsf{u}}[n]$ denote the number of offloading bits from GU k to the associated UAV over the time slot n. Besides, we denote the UAV computing resource allocated to handle the sub-task offloaded from GU k at time slot n by $f_k^{\mathsf{u}}[n]$ (CPU cycles/second). Then, the task execution time at the associated UAV in time slot n can be computed as $T_k^{\mathsf{edc}}[n]{=}\frac{l_k^{\mathsf{u}}[n]c_k}{f_k^{\mathsf{u}}[n]}$. Moreover, the energy consumption for executing the offloaded sub-task from GU k at the associated UAV in time slot n can be calculated as $E_k^{\mathsf{edc}}[n] = \kappa l_k^{\mathsf{u}}[n]c_k(f_k^{\mathsf{u}}[n])^2$. Hence, total energy consumption at the associated UAVs to process the offloading sub-task from GU k can be calculated as

$$E_k^{\mathsf{ed}} = \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]\big(\kappa l_k^{\mathsf{u}}[n]c_k(f_k^{\mathsf{u}}[n])^2\big). \tag{3}$$

In addition, we assume that the communication links from GUs to UAVs are dominated by the line-of-sight (LoS) propagation where the channel gains are mostly dependent on the UAV-GU distance. The distance between GU k and UAV m in time slot n can be calculated as $d_{k,m}[n]{=}\sqrt{H^2{+}\,\|\mathbf{q}_m[n] - \mathbf{r}_k^{\mathsf{u}}\|^2}$. Moreover, the channel power gain from GU k to UAV m in time slot n is assumed to follow the free-space path loss model and it can be expressed as $g_{k,m}[n]{=}\rho_0(d_{k,m}[n])^{-2}{=}\frac{\rho_0}{H^2+\left\|\mathbf{q}_m[n]-\mathbf{r}_k^{\mathsf{u}}\right\|^2}$, where ρ_0 presents the channel power gain at the reference distance of 1 m. Then, the achievable rate of the uplink transmission from GU k to the associated UAV m in time slot n denoted by $R_{k,m}^{\mathsf{u}}[n]$ in bits/second (bps), can be expressed as

$$R_{k,m}^{\mathsf{u}}[n] = \beta_k^{\mathsf{u}}[n]\log_2\left(1 + \frac{P_k^{\mathsf{u}}g_{k,m}[n]}{\beta_k^{\mathsf{u}}[n]\sigma^2}\right), \tag{4}$$

where $\beta_k^{\mathsf{u}}[n]$ and P_k^{u} represent the bandwidth allocated to GU k in time slot n and the transmit power of GU k for its uplink transmission, respectively, and σ^2 denotes the power density of the additive white Gaussian noise (AWGN) at the receiver. Then, the transmission time from GU k to UAV m in time slot n for the data related to the underlying offloaded sub-task can be expressed as $T_{k,m}^{\mathsf{edt}}[n] = \frac{l_k^{\mathsf{u}}[n]}{R_{k,m}^{\mathsf{u}}[n]}$. The energy consumption for the data transmission from GU k to UAV m in time slot n can be calculated as

$$E_{k,m}^{\mathsf{edt}}[n] = \frac{l_k^{\mathsf{u}}[n]P_k^{\mathsf{u}}}{R_{k,m}^{\mathsf{u}}[n]}. \tag{5}$$

Moreover, we assume that the partial task from GU k is offloaded and processed completely at each associated UAV in each time slot. Then we have following constraints

$$T_{k,m}^{\mathsf{ed}}[n]{=}\phi_{k,m}^{\mathsf{u}}[n]\left(\frac{l_k^{\mathsf{u}}[n]c_k}{f_k^{\mathsf{u}}[n]}{+}\frac{l_k^{\mathsf{u}}[n]}{R_{k,m}^{\mathsf{u}}[n]}\right) \le \Delta t, \forall k,m,n. \tag{6}$$

The total processing time at the UAVs to serve GU k can be

$$T_k^{\mathsf{ed}} = \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\theta_k[n]T_{k,m}^{\mathsf{ed}}[n]. \tag{7}$$

Furthermore, to remain in the air, the UAV will consume some energy during hovering time. Hence, the flying energy consumption of UAV m can be expressed as $E_m^{\mathsf{edf}} = P_m^{\mathsf{f}}T$, where P_m^{f} denotes the flying power of UAV m.

3) Satellite Cloud Computing Model: Because the cloud server has abundant computation and energy resources, we omit the processing time at the cloud server and the cloud energy consumption involved in task execution and transmission of the computation results from the cloud server to GUs. Assuming that the cloud server is located far from the considered terrestrial network area, we typically need a multi-hop satellite communication path to transmit the data related to the offloaded sub-task from each GU to the cloud server. The authors of the work [8] have proposed an algorithm to determine the number of hops (i.e., the number of inter-satellite links (ISL)) and the corresponding satellites to establish the multi-hop communication path between two locations on the ground (i.e., see Algorithm 1 of [8]). By employing this algorithm, we can determine the set of satellites involved in the transmission of the offloaded sub-task data and we denote this satellite set by $\mathcal{S}' \subset \mathcal{S}$. The number of hop counts between the first and the last satellites connecting the considered terrestrial network area and the cloud server can be calculated as $L = |\mathcal{S}'| - 1$, where $|\mathcal{S}'|$ is the number of satellites in the set $\mathcal{S}'$. Because of the large coverage radius of each satellite (i.e., coverage radius of each Starlink satellite is 580 km [9]), we assume that all GUs transmit their data to the first satellite in $\mathcal{S}'$ while the last satellite in $\mathcal{S}'$ can be directly connected to the ground cloud server. Hence, the total data transmission time and the propagation time from GU k to the cloud server can be calculated as

$$T_k^{\mathsf{cl}} = (1{-}\lambda_k^{\mathsf{lo}}{-}\lambda_k^{\mathsf{ed}})s_k\left(\frac{1}{R_k^{\mathsf{s}}}+\sum_{i=1}^{L}(\frac{1}{R_i^{\mathsf{ss}}})+\frac{1}{R^{\mathsf{cl}}}\right){+}T_k^{\mathsf{prop}}, \tag{8}$$

where $R_k^{\mathsf{s}}, R_i^{\mathsf{ss}}, R^{\mathsf{cl}}$ represent the transmission rates between GU k and the first satellite, between the satellites in the i-th hop, and between the last satellite and the cloud server, respectively. Here, T_k^{prop} represents the total propagation delay from GU k to the first satellite, between satellites over the L ISLs, and from the last satellite to the cloud server. Moreover, the energy consumption of GU k for transmitting the data related to the offloaded sub-task to the first satellite can be calculated as

$$E_k^{\mathsf{s}} = \frac{(1-\lambda_k^{\mathsf{lo}}-\lambda_k^{\mathsf{ed}})s_kP_k^{\mathsf{s}}}{R_k^{\mathsf{s}}}, \tag{9}$$

where P_k^{s} represents transmission power of GU k to satellite.

E. Problem Formulation

Our design objective is to minimize the weighted energy consumption of all GUs and UAVs for all involved computation tasks which can be calculated as

$$E^{\mathsf{sum}} = \alpha_1\left(\sum_{k\in\mathcal{K}}E_k^{\mathsf{ed}} + \sum_{m\in\mathcal{M}}P_m^{\mathsf{f}}T\right) +$$
$$\alpha_2\sum_{k\in\mathcal{K}}\left(E_k^{\mathsf{lo}}{+}\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]E_{k,m}^{\mathsf{edt}}[n]{+}E_k^{\mathsf{s}}\right), \tag{10}$$

where $\alpha_1, \alpha_2{\in}[0,1]$ represents the weight factors of the energy consumption of UAVs and GUs, respectively, which can balance the energy consumption between the GUs and UAVs.

For convenience, we gather different decision variables as user scheduling $\mathbf{\Theta}=\{\theta_k[n], \forall k, n\}$, partial offloading control $\mathbf{\Lambda}=\{\lambda_k^{\mathsf{lo}}, \lambda_k^{\mathsf{ed}}, \forall k\}$, bit allocation $\mathbf{L}=\{l_k^{\mathsf{u}}[n], \forall k, n\}$, bandwidth allocation $\boldsymbol{\beta}=\{\beta_k^{\mathsf{u}}[n], \forall k, n\}$, computation resource allocation $\mathbf{F}=\{f_k^{\mathsf{u}}[n], \forall k, n\}$, and UAV trajectory control $\mathbf{Q}=\{\mathbf{q}_m[n], \forall m, n\}$. Our design aims to minimize the weighted energy consumption of the GUs and UAVs while satisfying maximum delay constraints of individual computation tasks. The optimization problem can be formulated as

$$\textbf{(P):} \min_{\mathbf{\Theta},\mathbf{\Lambda},\mathbf{L},\boldsymbol{\beta},\mathbf{F},\mathbf{Q}} \quad E^{\mathsf{sum}} \tag{11}$$

$$\text{s.t. } T_k^{\mathsf{lo}} \leq T_k^{\mathsf{max}}, \forall k, \tag{11a}$$

$$T_k^{\mathsf{cl}} \leq T_k^{\mathsf{max}}, \forall k, \tag{11b}$$

$$\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} \theta_k[n] T_{k,m}^{\mathsf{ed}}[n] \leq T_k^{\mathsf{max}}, \forall k, \tag{11c}$$

$$\theta_k[n] T_{k,m}^{\mathsf{ed}}[n] \leq \Delta t, \forall k, m, n, \tag{11d}$$

$$\sum_{m\in\mathcal{M}}\sum_{t=0}^{N_k-1} \theta_k[n+t]\phi_{k,m}^{\mathsf{u}}[n+t]=N_k, n\in\{1,...,N-N_k\}, \tag{11e}$$

$$\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}} \theta_k[n]\phi_{k,m}^{\mathsf{u}}[n] l_k^{\mathsf{u}}[n] = \lambda_k^{\mathsf{ed}} s_k, \forall k, \tag{11f}$$

$$\sum_{k\in\mathcal{K}} \theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]\beta_k^{\mathsf{u}}[n] \leq W_m^{\mathsf{u}}, \forall m, n, \tag{11g}$$

$$\sum_{k\in\mathcal{K}} \theta_k[n]\phi_{k,m}^{\mathsf{u}}[n] f_k^{\mathsf{u}}[n] \leq F_m^{\mathsf{max}}, \forall m, n, \tag{11h}$$

$$\mathbf{q}_m[1] = \mathbf{q}_m[N], \forall m, \tag{11i}$$

$$\|\mathbf{q}_m[n+1]-\mathbf{q}_m[n]\|^2 \leq S_{\max}^2, \forall m, n=1,...,N-1, \tag{11j}$$

$$\|\mathbf{q}_m[n]-\mathbf{q}_j[n]\|^2 \geq d_{\min}^2, \ \forall n, m, j \neq m, \tag{11k}$$

$$\theta_k[n] \in \{0,1\}, \forall k, n, \tag{11l}$$

$$0 \leq \lambda_k^{\mathsf{lo}}, \lambda_k^{\mathsf{ed}}, 1-\lambda_k^{\mathsf{lo}}-\lambda_k^{\mathsf{ed}} \leq 1, \forall k, \tag{11m}$$

$$\beta_k^{\mathsf{u}}[n], f_k^{\mathsf{u}}[n], l_k^{\mathsf{u}}[n] \geq 0, \forall k, n, \tag{11n}$$

where constraints (11a)-(11d) capture the delay requirements for the GUs. Constraints (11e) and (11l) describe the binary user scheduling constraints for the GUs served by the associated UAVs. Constraints (11g) capture the bandwidth allocation for transmissions between the GUs and UAVs while constraints (11h) present the UAVs' computation constraints where F_m^{max} denotes the maximum computation resource of UAV m. It can be seen that the objective and constraint functions (11a)-(11d) are non-linear and integer decision variables are involved in (11l) for the user scheduling. Hence, problem (11) is a non-convex mixed integer non-linear optimization problem (MINLP), which is difficult to solve optimally. In the following section, we describe how to compute a good feasible solution for this problem.

III. Proposed Algorithm

We adopt the alternating optimization approach to solve problem (11) where we iteratively optimize each set of variables given the values of other variables in the corresponding sub-problems until convergence. We describe how to solve these different sub-problems in the following.

A. Optimization of User Scheduling

Given $\{\mathbf{L}, \mathbf{\Lambda}, \mathbf{F}, \boldsymbol{\beta}, \mathbf{Q}\}$, the problem of optimizing user scheduling $\mathbf{\Theta}$ can be formulated as

$$\textbf{(P1):} \min_{\mathbf{\Theta}} \ E^{\mathsf{sum}} \tag{12}$$

$$\text{s.t. constraints (11c)} - \text{(11h)}, \text{(11l)}.$$

It can be verified that problem (12) is a standard mixed integer linear program (MILP), which can be solved efficiently by using the CVX-Gurobi solver.

B. Optimization of Partial Offloading Control and Bit Allocation Over Time Slots

Given $\{\mathbf{\Theta}, \mathbf{F}, \boldsymbol{\beta}, \mathbf{Q}\}$, the problem of optimizing the partial offloading control and bit allocation $\{\mathbf{\Lambda}, \mathbf{L}\}$ can be formulated as

$$\textbf{(P2):} \ \min_{\mathbf{\Lambda},\mathbf{L}} \ E^{\mathsf{sum}} \tag{13}$$

$$\text{s.t. constraints (11a)} - \text{(11d)}, \text{(11f)}, \text{(11m)}, \text{(11n)}.$$

It can be verified that problem (13) is a linear program (LP), which can be solved by using the CVX-Gurobi solver.

C. Optimization of Computation Resource Allocation

Given $\{\mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{L}, \boldsymbol{\beta}, \mathbf{Q}\}$, the problem of optimizing UAVs' computation resource allocation $\mathbf{F}$ can be stated as

$$\textbf{(P3):} \ \min_{\mathbf{F}} \ \alpha_1\Big(\sum_{k,m,n}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]\big(\kappa l_k^{\mathsf{u}}[n]c_k(f_k^{\mathsf{u}}[n])^2\big)\Big) + E^{\mathsf{sumf}} \tag{14}$$

$$\text{s.t. constraints (11c)}, \text{(11d)}, \text{(11h)}, \text{(11n)},$$

where $E^{\mathsf{sumf}}=\alpha_2\Big(\sum_{k\in\mathcal{K}}\big(\kappa\lambda_k^{\mathsf{lo}}s_k c_k(f_k)^2+\frac{(1-\lambda_k^{\mathsf{lo}}-\lambda_k^{\mathsf{ed}})s_k P_k^{\mathsf{s}}}{R_k^{\mathsf{s}}}\big) + \sum_{k,m,n}\big(\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n] l_k^{\mathsf{u}}[n] P_k^{\mathsf{u}}\frac{1}{R_{k,m}^{\mathsf{u}}[n]}\big)\Big)+\alpha_1\sum_m P_m^{\mathsf{f}} T$.

It can be shown that $\frac{1}{f_k^{\mathsf{u}}[n]}$ and $(f_k^{\mathsf{u}}[n])^2$ are convex functions with respect to $f_k^{\mathsf{u}}[n]$. Hence, the objective function is a convex function and all constraints are linear. Therefore, problem (14) is a convex problem, which can be solved effectively by using the CVX-Gurobi solver.

D. Optimization of Bandwidth Allocation

For the given $\{\mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{L}, \mathbf{F}, \mathbf{Q}\}$, the problem of optimizing bandwidth allocation $\boldsymbol{\beta}$ can be formulated as

$$\textbf{(P4):} \ \min_{\boldsymbol{\beta}} \ \alpha_2\Big(\sum_{k,m,n}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n] l_k^{\mathsf{u}}[n] P_k^{\mathsf{u}}\frac{1}{R_{k,m}^{\mathsf{u}}[n]}\Big) + E^{\mathsf{sum1}} \tag{15}$$

$$\text{s.t. constraints (11c)}, \text{(11d)}, \text{(11g)}, \text{(11n)},$$

where $E^{\mathsf{sum1}}=\alpha_2\big(\sum_{k\in\mathcal{K}}\big(\kappa\lambda_k^{\mathsf{lo}}s_k c_k(f_k)^2+\frac{(1-\lambda_k^{\mathsf{lo}}-\lambda_k^{\mathsf{ed}})s_k P_k^{\mathsf{s}}}{R_k^{\mathsf{s}}}\big)\big) +\alpha_1\big(\sum_{k,m,n}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]\big(\kappa l_k^{\mathsf{u}}[n]c_k(f_k^{\mathsf{u}}[n])^2\big)+\sum_m P_m^{\mathsf{f}} T\big)$.

We first introduce auxiliary variables $\xi_{k,m}[n] = R_{k,m}^{\mathsf{u}}[n]=\beta_k^{\mathsf{u}}[n]\log_2\big(1+\frac{B_{k,m}[n]}{\beta_k^{\mathsf{u}}[n]}\big)$, where $B_{k,m}[n]=\frac{P_k^{\mathsf{u}} g_{k,m}[n]}{\sigma^2}$. It can be verified that $\beta_k^{\mathsf{u}}[n]\log_2\big(1+\frac{B_{k,m}[n]}{\beta_k^{\mathsf{u}}[n]}\big)$ is a concave function with respect to $\beta_k^{\mathsf{u}}[n]$. Using the successive convex approximation (SCA) method, the upper-bound for this concave function derived by using the first-order Taylor expansion at the given point $\beta_k^{\mathsf{u},r}[n]$ in the r-th iteration of the approximation process can be expressed in the following.

We have

$$\beta_k^{\mathsf{u}}[n]\log_2\left(1+\frac{B_{k,m}[n]}{\beta_k^{\mathsf{u}}[n]}\right)\leq\beta_k^{\mathsf{u},r}[n]\log_2\left(1+\frac{B_{k,m}[n]}{\beta_k^{\mathsf{u},r}[n]}\right)$$
$$+\left(\log_2\left(1+\frac{B_{k,m}[n]}{\beta_k^{\mathsf{u},r}[n]}\right)-\frac{\log_2(e)B_{k,m}[n]}{B_{k,m}[n]+\beta_k^{\mathsf{u},r}[n]}\right)(\beta_k^{\mathsf{u}}[n]-\beta_k^{\mathsf{u},r}[n])$$
$$\triangleq R_{k,m}^{\mathsf{ub}}[n]. \quad (16)$$

Then, problem (15) can be approximated as

$$\textbf{(P4'):}\ \min_{\boldsymbol{\beta},\boldsymbol{\Xi}}\quad\alpha_2\left(\sum_{k,m,n}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]l_k^{\mathsf{u}}[n]P_k^{\mathsf{u}}\frac{1}{\xi_{k,m}[n]}\right)+E^{\mathsf{sum1}} \quad (17)$$

$$\text{s.t.}\sum_{m,n}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]\left(\frac{l_k^{\mathsf{u}}[n]c_k}{f_k^{\mathsf{u}}[n]}+\frac{l_k^{\mathsf{u}}[n]}{\xi_{k,m}[n]}\right)\leq T_k^{\mathsf{max}},\forall k, \quad (17a)$$

$$\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]\left(\frac{l_k^{\mathsf{u}}[n]c_k}{f_k^{\mathsf{u}}[n]}+\frac{l_k^{\mathsf{u}}[n]}{\xi_{k,m}[n]}\right)\leq\Delta t,\forall k,m,n, \quad (17b)$$

$$\xi_{k,m}[n]\leq R_{k,m}^{\mathsf{ub}}[n],\forall k,m,n, \quad (17c)$$

constraints (11g), (11n),

where $\boldsymbol{\Xi}=\{\xi_{k,m}[n],\forall k,m,n\}$.

Since $\frac{1}{\xi_{k,m}[n]}$ is convex with respect to $\xi_{k,m}[n]$, it can be seen that the objective function is convex and all constraints are linear. Hence, problem (17) is a convex problem, which can be solved effectively by using the CVX-Gurobi solver.

E. Optimization of Multi-UAV Trajectory Control

Given $\{\boldsymbol{\Theta},\boldsymbol{\Lambda},\mathbf{L},\mathbf{F},\boldsymbol{\beta}\}$, the problem of optimizing multi-UAV trajectory control $\mathbf{Q}$ can be formulated as

$$\textbf{(P5):}\ \min_{\mathbf{Q}}\quad\alpha_2\left(\sum_{k,m,n}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]l_k^{\mathsf{u}}[n]P_k^{\mathsf{u}}\frac{1}{R_{k,m}^{\mathsf{u}}[n]}\right)+E^{\mathsf{sum1}} \quad (18)$$

s.t. constraints (11c), (11d), (11i), (11j), (11k).

To approximate this problem, we introduce auxiliary variables $\gamma_{k,m}[n]=R_{k,m}^{\mathsf{u}}[n]$ and $S_{k,m}[n]\leq H^2+\|\mathbf{q}_m[n]-\mathbf{r}_k^{\mathsf{u}}\|^2$ and we have

$$\gamma_{k,m}[n]\leq\beta_k^{\mathsf{u}}[n]\log_2\left(1+\frac{P_k^{\mathsf{u}}\rho_0}{\beta_k^{\mathsf{u}}[n]\sigma^2S_{k,m}[n]}\right). \quad (19)$$

It can be verified that $\beta_k^{\mathsf{u}}[n]\log_2\left(1+\frac{R_k[n]}{S_{k,m}[n]}\right)$ is convex with respect to $S_{k,m}[n]$, where $R_k[n]=\frac{P_k^{\mathsf{u}}\rho_0}{\beta_k^{\mathsf{u}}[n]\sigma^2}$. By applying the SCA method, the lower-bound for the right-hand-side (RHS) of (19) by using the first-order Taylor expansion at the given point $S_{k,m}^r[n]$ in the r-th iteration of the approximation process can be derived as

$$\beta_k^{\mathsf{u}}[n]\log_2\left(1+\frac{R_k[n]}{S_{k,m}[n]}\right)\geq\beta_k^{\mathsf{u}}[n]\Bigg(\log_2\left(S_{k,m}[n]+R_k[n]\right)$$
$$-\log_2(S_{k,m}^r[n])-\frac{\log_2(e)}{S_{k,m}^r[n]}(S_{k,m}[n]-S_{k,m}^r[n])\Bigg)\triangleq R_{k,m}^{\mathsf{lb}}[n]. \quad (20)$$

In addition, since $\|\mathbf{q}_m[n]-\mathbf{r}_k^{\mathsf{u}}\|^2$ is convex with respect to $\mathbf{q}_m[n]$, we have the following inequality by applying the first-order Taylor expansion at the given point $\mathbf{q}_m^r[n]$: $\|\mathbf{q}_m[n]-\mathbf{r}_k^{\mathsf{u}}\|^2\geq\|\mathbf{q}_m^r[n]-\mathbf{r}_k^{\mathsf{u}}\|^2+2\left(\mathbf{q}_m^r[n]-\mathbf{r}_k^{\mathsf{u}}\right)^T\left(\mathbf{q}_m[n]-\mathbf{q}_m^r[n]\right)$.

Furthermore, by applying the first-order Taylor expansion at the given point $\mathbf{q}_m^r[n]$ and $\mathbf{q}_j^r[n]$ to $\|\mathbf{q}_m[n]-\mathbf{q}_j[n]\|^2$, the left-hand-side (LHS) of constraints (11k) can be approximated as $\|\mathbf{q}_m[n]-\mathbf{q}_j[n]\|^2\geq-\left\|\mathbf{q}_m^r[n]-\mathbf{q}_j^r[n]\right\|^2+2\left(\mathbf{q}_m^r[n]-\mathbf{q}_j^r[n]\right)^T\left(\mathbf{q}_m[n]-\mathbf{q}_j[n]\right),\forall j\neq m,n$.

Therefore, problem (18) can be approximated by the following problem:

$$\textbf{(P5'):}\ \min_{\mathbf{Q},\boldsymbol{\Gamma},\mathbf{S}}\quad\alpha_2\left(\sum_{k,m,n}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]l_k^{\mathsf{u}}[n]P_k^{\mathsf{u}}\frac{1}{\gamma_{k,m}[n]}\right)+E^{\mathsf{sum1}} \quad (21)$$

$$\text{s.t.}\sum_{m,n}\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]\left(\frac{l_k^{\mathsf{u}}[n]c_k}{f_k^{\mathsf{u}}[n]}+\frac{l_k^{\mathsf{u}}[n]}{\gamma_{k,m}[n]}\right)\leq T_k^{\mathsf{max}},\forall k, \quad (21a)$$

$$\theta_k[n]\phi_{k,m}^{\mathsf{u}}[n]\left(\frac{l_k^{\mathsf{u}}[n]c_k}{f_k^{\mathsf{u}}[n]}+\frac{l_k^{\mathsf{u}}[n]}{\gamma_{k,m}[n]}\right)\leq\Delta t,\forall k,m,n \quad (21b)$$

$$\gamma_{k,m}[n]\leq R_{k,m}^{\mathsf{lb}}[n],\forall k,m,n, \quad (21c)$$

$$S_{k,m}[n]\leq\|\mathbf{q}_m^r[n]-\mathbf{r}_k^{\mathsf{u}}\|^2+$$
$$+2\left(\mathbf{q}_m^r[n]-\mathbf{r}_k^{\mathsf{u}}\right)^T\left(\mathbf{q}_m[n]-\mathbf{q}_m^r[n]\right)+H^2,\forall k,m,n, \quad (21d)$$

$$d_{\min}^2\leq-\left\|\mathbf{q}_m^r[n]-\mathbf{q}_j^r[n]\right\|^2+$$
$$+2\left(\mathbf{q}_m^r[n]-\mathbf{q}_j^r[n]\right)^T\left(\mathbf{q}_m[n]-\mathbf{q}_j[n]\right),\forall j\neq m,n, \quad (21e)$$

constraints (11i), (11j),

where $\boldsymbol{\Gamma}=\{\gamma_{k,m}[n],\forall k,m,n\}$, $\mathbf{S}=\{S_{k,m}[n],\forall k,m,n\}$.

Since $\frac{1}{\gamma_{k,m}[n]}$ is convex with respect to $\gamma_{k,m}[n]$, the objective function is convex and all constraints are linear. Therefore, problem (21) is a convex problem, which can be solved effectively by using the CVX-Gurobi solver. Using the results above, our proposed algorithm based on the alternating optimization method is described in Algorithm 1.

Algorithm 1 Integrated User Scheduling, Computation Offloading and Bit Allocation, Resource Allocation, and Multi-UAV Trajectory Control

1: Inputs: M, K, W, T, and locations of GUs, satellites and cloud server;
2: Objective and initialization: Min weighted energy consumption (E^{sum}); Let $r=1$;
3: Find number of hop-count L by running the Alg. 1 in [8];
4: Determine the bandwidth W_m^{u} (i.e., $\sum_{m\in\mathcal{M}}W_m^{\mathsf{u}}\leq W$);
5: **repeat**
6: Solve sub-problem (12) to obtain $\boldsymbol{\Theta}^r$;
7: Solve sub-problem (13) to obtain $\mathbf{L}^r$ and $\boldsymbol{\Lambda}^r$;
8: Solve sub-problem (14) to obtain $\mathbf{F}^r$;
9: Solve sub-problem (17) to obtain $\boldsymbol{\beta}^r$;
10: Solve sub-problem (21) to obtain $\mathbf{Q}^r$;
11: Update $r=r+1$;
12: **until** Convergence
13: Return $E^{\mathsf{sum},*}$, $\boldsymbol{\Theta}^*$, $\mathbf{L}^*$, $\boldsymbol{\Lambda}^*$, $\mathbf{F}^*$, $\boldsymbol{\beta}^*$, $\mathbf{Q}^*$.

IV. Numerical Result

We consider a simulation scenario with the cloud server located far from the considered network area: the GUs are located in Montreal (45.50°N, 73.56°W) and the cloud server is deployed in Vancouver (49.28°N, 123.12°W). By running the Alg. 1 in [8], we can determine the number of satellite hop counts $L=4$. The parameter setting for our simulations is similar to that in [5], [7] and values of key parameters are summarized in Table I. We assume that 10 GUs are located on the ground and their task size values are set as $s_k=[6,10,5,4,3,5,6,8,10,10]$ Mbits and their corresponding maximum delay is set as $T_k^{\mathsf{max}}=[2,3,2,2,1,3,2,3,3,3]$

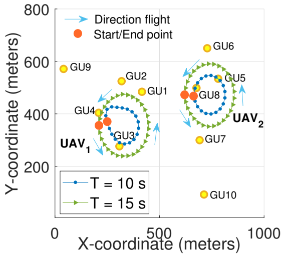

Fig. 2: Optimized trajectory with different UAV flight periods.

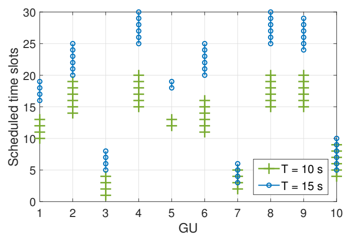

Fig. 3: User scheduling during UAV flight periods.

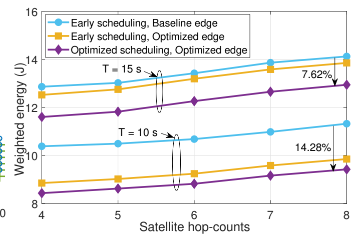

Fig. 4: Weighted sum of energy for different number of hops.

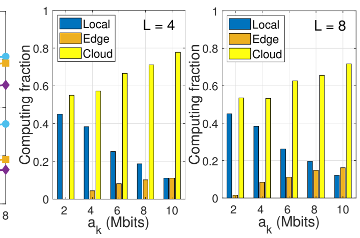

Fig. 5: Computation distribution for different task size values.

TABLE I: Simulation Parameters

Parameter	Description	Value
M	Number of UAVs	2
K	Number of GUs	10
H	Altitude of UAVs	100 m
P_k^{u}	Transmit power of GU with UAV	24 dBm
P_k^{s}	Transmit power of GU with satellite	30 dBm
P_m^{f}	Flying power consumption of UAV	32 dBm
f_k	Computation resource of GU (CPU cycles/s)	2×10^8
F_m^{max}	Maximum computation resource of UAV (CPU cycles/s)	3×10^9
σ^2	Noise power	-174 dBm
κ	Effective switched capacitance	10^{-28}
R_k^{s}	Transmission rate from GU to satellite	10 Mbps
R_i^{ss}	Transmission rate of ISLs	100 Mbps
R^{cl}	Transmission rate from satellite to cloud server	10 Mbps
T_k^{prop}	Total propagation delay from GU to cloud server	100 ms

(seconds). Other parameters are set as $c_k = 300$ (CPU cycles/bit), $[\alpha_1, \alpha_2] = [0.2, 0.8]$, $W = 10$MHz, $V_{\mathsf{max}} = 50$(m/s), $\Delta t = 0.5$s, and $d_{\mathsf{min}} = 20$m.

To investigate the effectiveness of the proposed algorithm, we consider the following baselines. In an "early scheduling" baseline, all GUs are scheduled from the first time slot of the UAV flight period. In another strategy named as "baseline edge" we initially set circular UAVs' trajectories to serve corresponding groups of GUs with the largest circumference of $V_{\mathsf{max}}T$, the values of partial offloading control variables are randomly generated (i.e., $\lambda_k^{\mathsf{lo}}, \lambda_k^{\mathsf{ed}} \in [0, 0.5]$), and a uniform allocation of bit, computation resource, and bandwidth is applied, i.e., $l_k^{\mathsf{u}}[n] = \lambda_k^{\mathsf{ed}} s_k / N_k$, $f_k^{\mathsf{u}}[n] = M F_m^{\mathsf{max}} / K$, and $\beta_k^{\mathsf{u}}[n] = W/K$, respectively. For comparison, the "optimized edge" represents our proposed design where these variables are optimized.

Fig. 2 illustrates the optimized trajectories in the scenarios of 10 GUs, $T = [10, 15]$s, and $L = 4$. As T increases, the UAVs must follow longer trajectories to accommodate the longer flight period and the optimized trajectories are smoother. Fig. 3 presents the scheduled time slots for individual GUs during UAVs flight periods $T = [10, 15]$s and $L = 4$. It can be seen that GUs are mostly scheduled when the associated UAVs are closer to them because this enables to achieve higher transmission rate and satisfy the delay constraints of underlying computation tasks.

In Fig. 4, we show the weighted sum of energy for scenarios with $T = [10, 15]$s and different satellite hop counts L in the space network layer. It can be seen that the weighted sum of energy increases with the number of satellite hop counts. This is because larger satellite hop counts means longer transmission and propagation delay in the space network layer, which would force higher computation load to be processed at the GUs and edge servers. Moreover, the weighted sum of energy also increases and the proposed algorithm can save less energy compared to the energy required by the "early scheduling" and "baseline edge" baselines for the longer flight period T. Fig. 5 illustrates the computation load distribution over network layers. This figure shows that larger task size values lead to less computation load distributed at the GUs while larger satellite hop counts result in higher computation loads to be processed at the edge servers.

V. Conclusion

In this paper, we have studied the joint optimization of user scheduling, partial offloading control and bit allocation, computation resource and bandwidth allocation, and UAV trajectory control for the SAGIN. Numerical results have demonstrated the effectiveness of the proposed algorithm compared to the baselines and the strong effects of various system parameters such as the numbers of satellite hop counts as well as the data size values to computation load distribution and the energy consumption.

References

[1] X. Zhu and C. Jiang, "Integrated satellite-terrestrial networks toward 6G: Architectures, applications, and challenges," *IEEE Internet of Things J.*, vol. 9, no. 1, pp. 437–461, Jan. 2022.

[2] T. Darwish, G. K. Kurt, H. Yanikomeroglu, M. Bellemare, and G. Lamontagne, "LEO satellites in 5G and beyond networks: A review from a standardization perspective," 2021. [Online]. Available: https://arxiv.org/abs/2110.08654

[3] T. Ma, H. Zhou, B. Qian, N. Cheng, X. Shen, X. Chen, and B. Bai, "UAV-LEO integrated backbone: A ubiquitous data collection approach for B5G internet of remote things networks," *IEEE J. Select. Areas Commun.*, vol. 39, no. 11, pp. 3491–3505, Nov. 2021.

[4] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Select. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[5] S. Mao, S. He, and J. Wu, "Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3992–4002, Sept. 2021.

[6] M. Feng, M. Krunz, and W. Zhang, "Joint task partitioning and user association for latency minimization in mobile edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8108–8121, Aug. 2021.

[7] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 313–12 325, Dec. 2018.

[8] Q. Chen, G. Giambene, L. Yang, C. Fan, and X. Chen, "Analysis of inter-satellite link paths for LEO mega-constellation networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2743–2755, Mar. 2021.

[9] S. Cakaj, "The parameters comparison of the "Starlink" LEO satellites constellation for different orbital shells," *Frontiers Commun. Netw.*, vol. 2, p. 7, 2021.