Joint Power Allocation and Task Scheduling for Data Offloading in Non-Geostationary Orbit Satellite Networks

Lijun He, *Member, IEEE*, Ziye Jia, *Member, IEEE*, Juncheng Wang, *Member, IEEE*, Erick Lansard, Zhu Han, *Fellow, IEEE*, and Chau Yuen, *Fellow, IEEE*

***Abstract*—In Non-Geostationary Orbit Satellite Networks (NGOSNs) with a large number of battery-carrying satellites, proper power allocation and task scheduling are crucial to improving data offloading efficiency. In this work, we jointly optimize power allocation and task scheduling to achieve energy-efficient data offloading in NGOSNs. Our goal is to properly balance the minimization of the total energy consumption and the maximization of the sum weights of tasks. Due to the tight coupling between power allocation and task scheduling, we first derive the optimal power allocation solution to the joint optimization problem with any given task scheduling policy. We then leverage the conflict graph model to transform the joint optimization problem into an Integer Linear Programming (ILP) problem with any given power allocation strategy. We explore the unique structure of the ILP problem to derive an efficient semidefinite relaxation-based solution. Finally, we utilize the genetic framework to combine the above special solutions as a two-layer solution for the original joint optimization problem. Simulation results demonstrate that our proposed solution can properly balance the reduction of total energy consumption and the improvement of the sum weights of tasks, thus achieving superior system performance over the current literature.**

***Index Terms*—Non-geostationary orbit satellite networks, data offloading, power allocation, task scheduling.**

Received 9 June 2024; revised 23 February 2025; accepted 13 April 2025. Date of publication 16 April 2025; date of current version 9 June 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62201463 and 62301251, in part by the Natural Science Foundation of Jiangsu Province of China under Project BK20220883, and in part by NSF ECCS-2302469, CMMI-2222810, Toyota. Amazon and Japan Science and Technology Agency (JST) Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE) JPMJAP2326. The associate editor coordinating the review of this article and approving it for publication was P. Calyam. *(Corresponding author: Ziye Jia.)*

Lijun He is with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: lijunhe@cumt.edu.cn).

Ziye Jia is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China, and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 211111, China (e-mail: jiaziye@nuaa.edu.cn).

Juncheng Wang is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, China (e-mail: jcwang@hkbu.edu.hk).

Erick Lansard and Chau Yuen are with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: erick.lansard@ntu.edu.sg; chau.yuen@ntu.edu.sg).

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/TNSM.2025.3561266

I. Introduction

DRIVEN by the fast proliferation of wireless communication applications, wireless connections among humans, machines, and things can happen everywhere [1]. As predicted in the Cisco research report [2], there will be over 500 billion connected devices by the end of 2030. It is challenging for such a huge number of devices to access terrestrial networks due to high network load and limited network coverage. To alleviate the network load and enhance the network coverage in terrestrial networks, Non-Geostationary Orbit Satellite Networks (NGOSNs) have emerged as a promising solution [3]. The huge backhaul capacity allows NGOSNs to support vast network connections. Furthermore, the top-down nature allows NGOSNs to provide global coverage with much lower costs than terrestrial networks [4].

In light of the benefits provided by NGOSNs, many commercial projects have started to deploy NGOSNs in space for global coverage, such as Starlink, Kuiper, SpaceMobile, and Telesat [5], [6], [7]. These commercial projects plan to launch a large number of satellites into Low Earth Orbit (LEO) to build their individual NGOSNs. The fast proliferation of LEO satellites in NGOSNs inevitably results in an explosion of space data [8]. Up to date, NGOSNs mainly rely on the global arrangement of Ground Stations (GSs) to receive these space data. Although the communication capacity of Satellite-Ground Links (SGLs) is large, NGOSNs are essentially line-of-sight communication networks [9]. Due to the rapid orbital motion of LEO satellites, the data offloading in NGOSNs can only occur within limited Transmission Time Windows (TTWs) between the LEO satellites and the GSs. As such, the data offloading capability of NGOSNs is greatly limited by the TTWs.

One key factor for efficient data offloading in NGOSNs is to properly schedule the data offloading tasks within limited TTWs, in order to maximize the desired network performance, such as the sum weights of tasks. The explosion of space data also brings the concern of high energy consumption in NGOSNs [10]. It is of great importance to offload the space data in NGOSNs with as little energy as possible due to three fundamental drivers: First, the LEO satellites in NGOSNs are powered by rechargable batteries, which can only be charged through the solar panels when the satellites are at the sunny side of the earth. Second, the lifetime of LEO satellites can be prolonged through efficient energy consumption control to

reduce the number of battery charge-discharge cycles. Third, minimizing energy consumption can reduce the number of batteries needed in LEO satellites, reducing the size of the satellite and its operational costs. Therefore, proper power allocation to minimize the energy consumption is another key factor for efficient data offloading in NGOSNs.

In brief, joint consideration of power allocation and task scheduling is critical, to properly balance the improvement of desired network performance and the reduction of total energy consumption, for efficient data offloading in NGOSNs. This problem is challenging mainly due to the following reasons:

1) *Intermittence and heterogeneity of SGLs*. High-speed motion of LEO satellites leads to intermittent SGLs, which can be modeled as a set of time windows. The lengths of the SGL time windows vary with satellite orbit parameters and can change over time. Furthermore, the transmission time for each data offloading task depends on the amount of data to be offloaded and the transmission rate that varies with the SGL time windows. Therefore, it is challenging to efficiently match the data offloading tasks and the short SGL time windows without causing resource conflict.

2) *Tight coupling between power allocation and task scheduling*. Both power allocation and task scheduling affects the total energy consumption as well as the desired network performance. It is thus challenging to decouple power allocation and task scheduling and make joint decisions to properly tradeoff the improvement of desired network performance and the reduction of total energy consumption in NGOSNs.

To address the above challenges, in this work, we jointly optimize power allocation and task scheduling in NGOSNs under the time window constraints, to balance the two conflicting objectives of sum task weight maximization and total energy consumption minimization. The main contributions of this paper are summarized as follows:

- We put forward a joint optimization framework that incorporates power allocation and task scheduling into data offloading, to reveal the interplay between the total energy consumption and the sum task weight in NGOSNs. Our goal is to minimize the normalized weighted sum of the total energy consumption and the sum task weight, under practical constraints such as maximum power range, minimum transmission rate, and time window limits. The resulting joint power allocation and task scheduling problem captures the features of the data offloading in practical NGOSNs.
- To efficiently solve the joint optimization problem, we first consider its two special cases: a Power Allocation (PA) problem with any given task scheduling policy and a Task Scheduling (TS) problem with any given power allocation strategy. Then, we derive an optimal solution to the PA problem with low computation complexity and utilize conflict graph to reformulate the TS problem into a special Integer Linear Programming (ILP) problem, which can be efficiently solved via Semi-Definite Relaxation (SDR). Finally, we embed the two special solutions into a genetic framework to provide a two-layer solution, termed as Energy-efficient Data Offloading (EDO), for the original joint optimization problem.
- We conduct extensive experiments on the NGOSNs simulation platform. Our simulation results demonstrate the following insights: 1) Reducing the feasible set of power allocation can substantially improve the system performance. 2) The proposed EDO algorithm efficiently trades off the two conflicting objectives of maximizing the sum task weight and minimizing the total energy consumption over a wide range of the tuning coefficient. 3) EDO substantially outperforms the state-of-the-art alternatives and its feasibility and convergence are demonstrated under various parameter settings.

The rest of this paper is organized as follows. In Section II, we introduce the related work. Section III describes the system model and problem formulation. In Section IV, we elaborate the proposed EDO algorithm for the joint power allocation and task scheduling problem. Simulation results and discussions are presented in Section V, followed by concluding remarks in Section VI.

II. Related Work

Data offloading in NGOSNs mainly focuses on how to efficiently allocate the limited space resources within certain time windows to satisfy the requirements of data offloading tasks. From the mathematical perspective, this problem is similar to the Parallel Machine Scheduling Problem with Time Windows (PMSPTW). Recently, a majority of works studied PMSPTW in satellite networks to design efficient task scheduling algorithms. These works can be divided into two categories.

The first category [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] focused on how to schedule Inter-Satellite Links (ISLs) between data relay satellites and LEO satellites to maximize the data offloading efficiency in Data Relay Satellite Networks (DRSNs). In terms of algorithm performance, these works can be further divided into exact algorithms[11], [12], heuristic algorithms[13], [14], [15], [16], [17], [18], [19], and meta-heuristic algorithms[20], [21]. Based on dynamic programming [11] and branch-and-bound method [12], the exact algorithms aim to find an optimal solution to the PMSPTW, resulting in high computational complexity. Therefore, they are only suitable for small problem sizes.

To accommodate the larger network scale of DRSNs, some heuristic algorithms [13], [14], [15], [16], [17], [18], [19], [20], [21] have been proposed with low computation complexity at the cost of performance loss. These heuristic algorithms utilize the characteristics of DRSNs to obtain some expert knowledge and then devise search rules to improve the solution, without any guarantee on the optimality of the solution. To be specific, [13] and [14] proposed conflict resolution strategies to optimize the starting times of scheduling tasks within the available time windows. In [15] and [16], task sequencing rules were studied to maximize the network performance of DRSNs. In [17] and [18], task-interpolation strategies were adopted to determine when to offload space data in DRSNs. The data offloading problem in DRSNs was transformed into a sequential decision-making problem and then solved via a Deep Reinforcement Learning (DRL) based task scheduling

TABLE I
COMPARISON OF EXISTING WORKS ON SGLS SCHEDULING

	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[33]	[34]	EDO
Polynomial complexity	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Variable-size tasks	✓	✓	×	✓	✓	✓	✓	✓	✓	×	×	×	×	✓
Power allocation	×	×	×	×	×	×	×	×	×	×	✓	✓	✓	✓
Task scheduling	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓
Intermittent SGLs	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	✓

method [19]. To further improve the data offloading efficiency of DRSNs, some meta-heuristics were developed to address the PMSPTW. For example, an evolutionary method was utilized to solve the satellite data download scheduling problem [20]. The graph structure of DRSNs was used to formulate the data offloading problem, which is then solved through a based-genetic evolution algorithm [21].

The second category [22], [23], [24], [25], [26], [27], [28], [29], [30] studied the PMSPTW to select a set of optimal GSs for LEO satellites to schedule SGLs while satisfying the time window constraints. The scheduling algorithms in these works are either deterministic [22], [23] or non-deterministic [24], [25], [26], [27], [28], [29], [30]. The deterministic algorithms use mathematical tools, such as tree search [22] and dynamic programming [23], to find an optimal solution under mild assumptions. However, the deterministic algorithms are time-consuming especially for large-scale problems, and thus are not suitable for practice NGOSNs. To accommodate large-scale NGOSNs, some non-deterministic algorithms have been proposed to find approximation solutions with lower computation costs. They are mainly based on some meta-heuristic methods, such as particle swarm optimization [24], ant colony optimization [25], and genetic algorithm [26], [27], [28]. Recently, some attempts have been made to utilize DRL for solving satellite scheduling problems. For example, the PMSPTW was transformed into an Assignment Problem (AP) and a Single Antenna Scheduling Problem (SASP) [29]. Then, the DRL method and a heuristic scheduling method were proposed to solve the AP and the SASP, respectively. In [30], a DRL-based algorithm was proposed for a rapid satellite range rescheduling problem to process real-time emergency events.

However, most works do not consider the energy consumption control of battery-carrying LEO satellites powered by solar panels, resulting in data offloading interruptions due to the overuse of energy. Energy consumption control is crucial for efficient data offloading in NGOSNs. To improve data offloading efficiency, some researchers studied energy-efficient data offloading in NGOSNs, such as land mobile satellite systems [31], cognitive satellite terrestrial networks [32], multi-cell satellite-terrestrial networks [33], and small satellite cluster networks [34]. In these works, each offloading task is represented as a set of equal packets, and then energy-efficient scheduling strategies are proposed to download these packets. In this case, the decision variables for scheduling packets form a huge solution space, increasing the difficulty of solving the problem. Therefore, these works [31], [32], [33], [34] are unsuitable for NGOSNs to offload the tasks with a large amount of data.

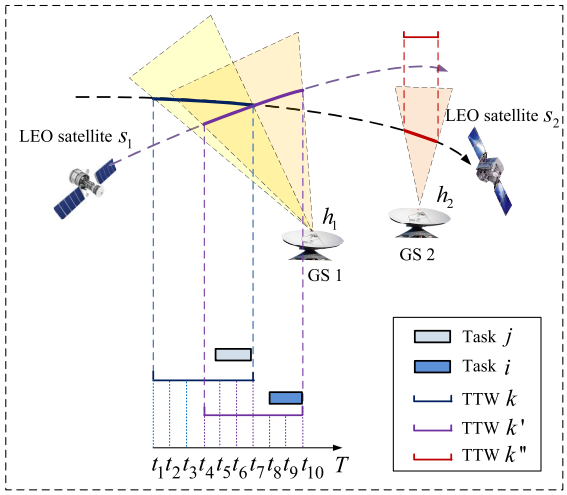


Fig. 1. An illustration of data offloading from the GSs to the LEO satellites in NGOSNs.

In contrast to previous work [31], [32], [33], [34], in this work, we characterize the main features of data offloading in real NGOSNs from the following three aspects: First, we consider a more realistic data offloading scenario with variable-size tasks. Second, our proposed joint optimization framework incorporates the constraints of the time window on data offloading into NGOSNs, which captures the intermittent characteristic of SGLs. Third, we develop an efficient yet fast two-layer optimization solution, the EDO algorithm, to simultaneously minimize total energy consumption and maximize the sum weights of tasks. The distinctions between existing works on SGLs scheduling and our work are succinctly summarized in Table I.

III. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a downlink NGOSN consisting of a set $\mathcal{S} = \{1, 2, \ldots, S\}$ of LEO satellites and a set of GSs. The whole scheduling horizon is segmented into T time slots represented by $\mathcal{T} = \{1, 2, \ldots, T\}$. We use time slot t to represent the time interval of $[t, t+1]$. Without loss of generality, each LEO satellite is equipped with one transmit antenna and each GS is equipped with one receiver antenna. Let $\mathcal{H} = \{1, 2, \ldots, H\}$ denote the set of data receiver antennas. As shown in Fig. 1, LEO satellites cycle their individual orbits at high speed, resulting in a set of short contact TTWs. Let $\mathcal{K}_{s,h}$ denote the set of TTWs between LEO satellite $s \in \mathcal{S}$ and antenna $h \in \mathcal{H}$. Let $\mathcal{K}_{s,\mathcal{H}} = \{\mathcal{K}_{s,h}, h \in \mathcal{H}\}$. We use 4-tuples $\{s, h, a_k, b_k\}$ to represent each TTW $k \in \mathcal{K}_{s,h}$, where a_k and b_k are the beginning and end times of TTW k, respectively. The set of

data offloading tasks is denoted by $\mathcal{J} = \{1, 2, \ldots, J\}$. Let $s(j)$ represent the LEO satellite that stores the data of task j.

A. SGL Communication Model

We use R_{jk}^{SGL} (bits/s) to represent the data rate of SGLs within TTW k for any task j, given by

$$R_{jk}^{\mathrm{SGL}} = B_c \log_2\left(1 + \gamma_{jk}\right), \ \forall j \in \mathcal{J}, k \in \mathcal{K}_{s(j),\mathcal{H}}, \quad (1)$$

where B_c is the bandwidth and γ_{jk} is the Signal-to-Noise Ratio (SNR) of offloading task j within TTW k. We evaluate the γ_{jk} of SGLs [35] as follows:

$$\gamma_{jk} = \frac{P_{s(j)} G_{s(j)}^{\mathrm{tran}} G_h^{\mathrm{rec}} L_f L_l^k}{N_0}, \quad (2)$$

where $P_{s(j)}$ is the transmit power of LEO satellite $s(j)$ allocated to task j, $G_{s(j)}^{\mathrm{tran}}$ is the transmit antenna gain of LEO satellite $s(j)$, G_h^{rec} is the receiver antenna gain of h, L_f is the free space loss, L_l^k is the total path loss for TTW k, and N_0 is the noise power. The total path loss in dB can be calculated as $L_l^k = L_b^k + L_g^k + L_s^k + L_e^k$, where L_b^k is the basic path loss, L_g^k represents the attenuation due to atmospheric gasses, L_s^k indicates the attenuation due to either ionospheric or tropospheric scintillation, and L_e^k is the building entry loss for TTW k [3]. Each LEO satellite $s(j)$ is subject to the following transmit power constraint:

$$\mathrm{C1}: 0 \le P_{s(j)} \le P_{s(j)}^{\max}, \ \forall j \in \mathcal{J}, \quad (3)$$

where $P_{s(j)}^{\max}$ is the maximum transmit power limit on LEO satellite $s(j)$. Furthermore, the data rate of SGLs in TTW k for task j is subject to the following minimum rate constraint:

$$\mathrm{C2}: R_{jk}^{\mathrm{SGL}} \ge R_k^{\min}, \ \forall j \in \mathcal{J}, k \in \mathcal{K}_{s(j),\mathcal{H}}, \quad (4)$$

where $R_k^{\min}$ is the minimum rate limit of TTW k.

B. Task Offloading Model

Let D_j and w_j denote the data amount and the weight of task j, respectively. We use p_{jk} to represent the processing time, i.e., the amount of time to transmit task j within TTW k, which can be computed by

$$p_{jk} = \frac{D_j}{R_{jk}^{\mathrm{SGL}}}. \quad (5)$$

Let st_j and et_j be the earliest time and the latest time to start offloading task j, respectively. We use 6-tuples $\{s(j), D_j, w_j, p_{jk}, st_j, et_j\}$ to represent each task $j \in \mathcal{J}$. Furthermore, we use a decision variable $x_{jk}^t \in \{0, 1\}$ to represent the offloading strategy, such that $x_{jk}^t = 1$ indicates that task j is offloaded in time slot t within TTW k; and $x_{jk}^t = 0$ otherwise. Thus, the offloading strategy is subject to the following binary constraint:

$$\mathrm{C3}: x_{jk}^t \in \{0, 1\}, \ \forall j \in \mathcal{J}, k \in \mathcal{K}_{s(j),\mathcal{H}}, t \in \mathcal{F}(j). \quad (6)$$

This constraint indicates whether task j is to be offloaded in time slot t within TTW k, or not.

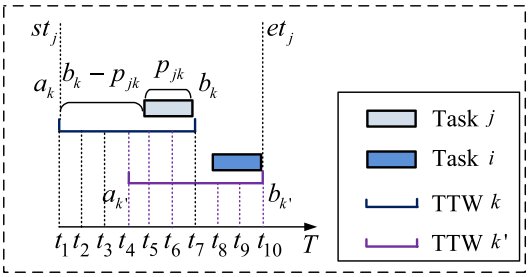


Fig. 2. An illustration of the set of feasible time slots $\mathcal{F}(j,k) = \{t_1, t_2, \ldots, t_5\}$ within TTW k.

We use $\mathcal{F}(j)$ to represent the set of all feasible time slots for task j, given by

$$\mathcal{F}(j) = \bigcup_{k \in \mathcal{K}_{s(j),\mathcal{H}}} \mathcal{F}(j,k), \quad (7)$$

where $\mathcal{F}(j,k) = [a_k, b_k - p_{jk}] \cap [st_j, et_j]$ is the set of feasible slots within TTW k to start offloading task j. In Fig. 2, we give an illustration of the set of feasible slots $\mathcal{F}(j,k) = \{t_1, t_2, \ldots, t_5\}$. We further introduce the following constraint to guarantee that each task j is offloaded at most once:

$$\mathrm{C4}: \sum_{t \in \mathcal{F}(j)} x_{jk}^t \le 1, \ \forall j \in \mathcal{J}. \quad (8)$$

In addition, we use E_j to represent the energy consumption for offloading task j, given by

$$E_j = \sum_{t \in \mathcal{F}(j)} x_{jk}^t P_{s(j)} p_{jk}. \quad (9)$$

C. Time Window Model

We define the set of occupying time slots for antenna h to receive task j in time slot t as follows:

$$\Phi(j,h,t) = \bigcup_{k \in \mathcal{K}_{s(j),h}} \left[t, t + p_{jk} - 1\right] \cap [a_k, b_k] \cap \left[st_j, et_j\right].$$

We can see from the example in Fig. 2 that $\Phi(j,h,t_5) = \{t_5, t_6\}$. We add the following constraint that any receiver antenna h receives at most one task in any time slot:

$$\mathrm{C5}: \sum_{j \in \mathcal{J}} \sum_{\tau \in \Phi(j,h,t)} x_{jk}^\tau \le 1, \ \forall h \in \mathcal{H}, t \in \mathcal{F}(j). \quad (10)$$

Similarly, we define the set of occupying feasible time slots for LEO satellite s to offload task j in time slot t as follows:

$$\Theta(j,s,t) = \bigcup_{k \in \mathcal{K}_{s(j),\mathcal{H}}} \left[t, t + p_{jk} - 1\right] \cap [a_k, b_k] \cap \left[st_j, et_j\right].$$

We further use the following constraint to indicate that any LEO satellite offloads at most one task in any time slot:

$$\mathrm{C6}: \sum_{j \in \mathcal{J}} \sum_{\tau \in \Theta(j,s,t)} x_{jk}^\tau \le 1, \ \forall s \in \mathcal{S}, t \in \mathcal{F}(j). \quad (11)$$

Our key notations are summarized in Table II.

TABLE II
SUMMARY OF KEY NOTATIONS

Notation	Description
$\mathcal{S}$	Set of LEO satellites
$\mathcal{T}$	Set of time slots
$\mathcal{H}$	Set of data receiver antennas equiped on GSs
$\mathcal{K}_{s,h}$	Set of TTWs between LEO satellite s and antenna h
a_k	Beginning time of TTW k
b_k	End time of TTW k
$\mathcal{J}$	Set of data offloading tasks
$s(j)$	LEO satellite that stores task j
R_{jk}^{SGL}	Achievable data rate of SGLs within TTW k for task j
γ_{jk}	SNR when offloading task j within TTW k
$G_{s(j)}^{\text{tran}}$	Transmit antenna gain of LEO satellite $s(j)$
G_h^{rec}	Receiver antenna gain of antenna h
L_f	Free space loss
L_l^k	Total path loss of TTW k
N_0	Noise power
$P_{s(j)}$	Transmit power of LEO satellite $s(j)$ allocated to task j
$P_{s(j)}^{\max}$	Maximum transmit power of LEO satellite $s(j)$
E_j	Energy consumption for offloading task j
$R_k^{\min}$	Minimum rate limit of TTW k
D_j	Data amount of task j
w_j	Weight of task j
p_{jk}	Transmission time of task j within TTW k
st_j	Earliest time to start offloading task j
et_j	Latest time to start offloading task j
$\mathcal{F}(j)$	Set of all feasible time slots for task j
$\mathcal{F}(j,k)$	Set of feasible slots within TTW k to start offloading task j
$\Phi(j,h,t)$	Set of occupying time slots for antenna h to receive task j in time slot t
$\Theta(j,s,t)$	Set of occupying feasible time slots for LEO satellite s to offload task j in time slot t

D. Joint Optimization Problem Formulation

To balance the two conflicting objectives of minimizing the total energy consumption and maximizing the sum task weight, we jointly optimize power allocation and task scheduling, to minimize their normalized weighted sum subject to the constraints introduced above. The resulting joint optimization problem is formulated as follows:

$$\mathbf{P0}(\boldsymbol{x},\boldsymbol{P}):\quad \min_{\boldsymbol{x},\boldsymbol{P}}\ \frac{\lambda}{E_{\max}^{\text{total}}}\sum_{j\in\mathcal{J}}E_j-\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{F}(j)}\frac{1-\lambda}{W_{\max}}w_j x_{jk}^t \quad \text{s.t. C1-C6,}$$

where $\boldsymbol{x}=\{x_{jk}^t\}$, $\boldsymbol{P}=\{P_{s(j)}\}$, $W_{\max}=\sum_{j\in\mathcal{J}}w_j$ is the sum weights of all the tasks, $E_{\max}^{\text{total}}$ is the sum of the maximization energy consumption of all tasks, and $\lambda\in[0,1)$ is a tuning coefficient to trade off the total energy consumption and the sum task weight. In particular, the weighted values represent significant system metrics, encompassing priorities and data volume, among others.

Theorem 1: Problem $\mathbf{P0}(\boldsymbol{P},\boldsymbol{x})$ is NP-hard in general.

Proof: By substituting $P_{s(j)}=P_{s(j)}^{\max}$, $\forall j\in\mathcal{J}$ into $\mathbf{P0}(\boldsymbol{P},\boldsymbol{x})$, we can derive its simplified version as follows:

$$\mathbf{P0}^{\text{simple}}(\boldsymbol{x}):\quad \max_{\boldsymbol{x}}\ \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{F}(j)}\left(\frac{1-\lambda}{W_{\max}}w_j-\frac{\lambda}{E_{\max}^{\text{total}}}F_{jk}\right)x_{jk}^t \quad \text{s.t. C3-C6,}$$

where $F_{jk}=\dfrac{P_{s(j)}^{\max}D_j}{B_c\log_2(1+\frac{P_{s(j)}^{\max}G_{s(j)}^{\text{tran}}G_h^{\text{rec}}L_f L_l^k}{N_0})}$. Problem $\mathbf{P0}^{\text{simple}}(\boldsymbol{x})$ is a typical satellite range scheduling problem, widely acknowledged as an NP-hard problem [36]. As such, we deduce that $\mathbf{P0}(\boldsymbol{P},\boldsymbol{x})$ is NP-hard. This completes the proof of Theorem 1. ■

Remark 1: In $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$, we use a normalized weighted sum method to balance the two conflicting objectives. This technique is a prevalent multi-objective optimization method that can obtain a complete pareto-optimal set for $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$, while also giving equal emphasis on each objective function[37]. In practical applications, the values of λ can be systematically adjusted within the interval [0, 1) to generate a set of feasible solutions for both total energy consumption and the sum weights of tasks, thus aiding in their selection. Specifically, the interval [0, 1) can be partitioned into *M* segments to produce *M* pairs of the total energy consumption and the sum weights of tasks. From these *M* alternatives, one can then choose the pair that most closely aligns with the desired criteria.

Remark 2: Note that the high dynamic characteristic of the data offloading in NGOSNs is mainly reflected in the time window constraints of C5 and C6. Furthermore, due to the constraints of C5 and C6, $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ falls into a kind of typical NP-hard PMSPTW and is thus challenging to solve. The tight coupling between $\boldsymbol{x}$ and $\boldsymbol{P}$ in the objective further complicates $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$. To circumvent these challenges, we draw insights on the structure of $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ to design our EDO algorithm.

IV. ENERGY-EFFICIENT DATA OFFLOADING

In this section, we first study two cases of $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ with any given task scheduling strategy and any given power allocation policy, to obtain two special solutions. We then utilize the genetic framework to combine the two special solutions as an efficient two-layer solution for the original joint optimization problem $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$.

A. Power Allocation for Fixed Task Scheduling

With any given task scheduling decision $\boldsymbol{x}$, $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ can be transformed into the following total energy consumption minimization problem:

$$\mathbf{P1}(\boldsymbol{P}):\quad \min_{\boldsymbol{P}}\ \frac{\lambda}{E_{\max}^{\text{total}}}\sum_{j\in\mathcal{J}}E_j-\Psi \quad \text{s.t. C1-C2,}$$

where Ψ is a constant, given by

$$\Psi=\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{F}(j)}\frac{1-\lambda}{W_{\max}}w_j x_{jk}^t. \tag{12}$$

We further explore the unique structure of $\mathbf{P1}(\boldsymbol{P})$ to obtain its optimal power allocation solution. We observe that $\mathbf{P1}(\boldsymbol{P})$ can be equivalently decomposed into *J* independent subproblems, one for each task *j*, given by

$$\mathbf{P2}(\boldsymbol{P}):\quad \min_{P_{s(j)}}\ \frac{\lambda}{E_{\max}^{\text{total}}}E_j \quad \text{s.t. C1-C2.}$$

We now solve $\mathbf{P2}(\boldsymbol{P})$. Combining (1) and (2), the data rate of SGLs within TTW k for any task j can be rewritten as

$$R_{jk}^{\mathrm{SGL}} = B_c \log_2\left(1+\frac{P_{s(j)}G_{s(j)}^{\mathrm{tran}}G_h^{\mathrm{rec}}L_f L_l^k}{N_0}\right), \quad \forall j\in\mathcal{J}, k\in\mathcal{K}_{s(j),\mathcal{H}}. \tag{13}$$

Also, from (5) and (9), the energy consumption for offloading task j can be expressed as

$$E_j = \sum_{t\in\mathcal{F}(j)} \frac{x_{jk}^t P_{s(j)} D_j}{R_{jk}^{\mathrm{SGL}}}, \ \forall j\in\mathcal{J}. \tag{14}$$

We then combine (13) and (14) to obtain:

$$E_j = \frac{\alpha P_{s(j)}}{\log_2\left(1+\beta P_{s(j)}\right)}, \ \forall j\in\mathcal{J}, \tag{15}$$

where $\alpha = \sum_{t\in\mathcal{F}(j)} \frac{x_{jk}^t D_j}{B_c}$ and $\beta = \frac{G_{s(j)}^{\mathrm{tran}}G_h^{\mathrm{rec}}L_f L_l^k}{N_0}$. To study the properties of E_j, we introduce a new variable $\boldsymbol{y} = \{y_j\}$ with each element defined as

$$y_j = \frac{1}{\log_2\left(1+\beta P_{s(j)}\right)}, \ \forall j\in\mathcal{J}. \tag{16}$$

We use $\boldsymbol{y}$ to equivalently transform $\mathbf{P2}(\boldsymbol{P})$ into the following new form:

$$\mathbf{P3}(\boldsymbol{y}): \ \min_{y_j} \ \frac{\alpha}{\beta} y_j\left(2^{\frac{1}{y_j}}-1\right) \quad \text{s.t.} \ \ f\left(P_{s(j)}^{\max}\right) \le y_j \le \frac{B_c}{R_k^{\min}},$$

where $f(P_{s(j)}^{\max}) = \frac{1}{\log_2(1+\beta P_{s(j)}^{\max})}$ and $g(y_j) = y_j(2^{\frac{1}{y_j}}-1)$.

We can obtain the first-order and second-order derivation of $g(y_j)$ in $\mathbf{P3}(\boldsymbol{y})$ as follows:

$$\nabla g(y_j) = 2^{\frac{1}{y_j}}\left(1-\frac{\ln 2}{y_j}\right)-1, \tag{17}$$

$$\nabla^2 g(y_j) = 2^{\frac{1}{y_j}}\frac{(\ln 2)^2}{(y_j)^3}. \tag{18}$$

From (17) and (18), we can see that $\forall y_j$ subject to $f(P_{s(j)}^{\max}) \le y_j \le \frac{B_c}{R_k^{\min}}$, we have $\nabla^2 g(y_j) > 0$, which indicates that $g(y_j)$ is a convex and unimodal function. Thus, the optimal solution to $\mathbf{P3}(\boldsymbol{y})$ is either $f(P_{s(j)}^{\max})$, $\frac{B_c}{R_k^{\min}}$, or an extreme point of $g(y_j)$, depending on the value of $\hat{y}_j$, given by

$$y_j^* = \begin{cases} f\left(P_{s(j)}^{\max}\right), & \text{if } \ \hat{y}_j < f\left(P_{s(j)}^{\max}\right), \\ \hat{y}_j, & \text{if } \ f\left(P_{s(j)}^{\max}\right) \le \hat{y}_j < \frac{B_c}{R_k^{\min}}, \\ \frac{B_c}{R_k^{\min}}, & \text{if } \ \frac{B_c}{R_k^{\min}} \le \hat{y}_j, \end{cases} \tag{19}$$

where $\hat{y}_j$ satisfies $\nabla g(\hat{y}_j) = 2^{\frac{1}{\hat{y}_j}}(1-\frac{\ln 2}{\hat{y}_j})-1 = 0$. We can use bisection search to obtain $\hat{y}_j, \forall j \in \mathcal{J}$. Therefore, we obtain an optimal solution to $\mathbf{P1}(\boldsymbol{P})$:

$$\boldsymbol{P}^* = \{P_{s(j)}^* | P_{s(j)}^* = \frac{1}{\beta}\left(2^{y_j^*}-1\right), \ \forall j\in\mathcal{J}\}. \tag{20}$$

Algorithm 1 Optimal Power Allocation Algorithm With Fixed Task Scheduling (OPA-FTS)

Input: $P_{s(j)}^{\max}$, B_c, $R_k^{\min}$, β.
for $j = 1{:}|\mathcal{J}|$ **do**
  $\llcorner y_j = \frac{1}{\log_2(1+\beta P_{s(j)}^{\max})}$, $\ulcorner y_j = \frac{B_c}{R_k^{\min}}$.
  while $\ulcorner y_j - \llcorner y_j > \epsilon$ **do**
    $\hat{y}_j = \frac{\ulcorner y_j + \llcorner y_j}{2}$.
    if $\nabla g(\hat{y}_j) \times \nabla g(\llcorner y_j) > 0$ **then**
      $\llcorner y_j = \hat{y}_j$.
    else
      $\ulcorner y_j = \hat{y}_j$.
    end if
  end while
  if $\hat{y}_j \le f(P_{s(j)}^{\max})$ **then**
    $y_j^* = f(P_{s(j)}^{\max})$.
  else
    if $f(P_{s(j)}^{\max}) < \hat{y}_j \le \frac{B_c}{R_k^{\min}}$ **then**
      $y_j^* = \hat{y}_j$.
    else
      $y_j^* = \frac{B_c}{R_k^{\min}}$.
    end if
  end if
  $P_{s(j)}^* = \frac{1}{\beta}(2^{y_j^*}-1)$.
end for
Output: $\boldsymbol{P}^* = \{P_{s(j)}^*\}$.

In Algorithm 1, we summarize the above solution, termed as Optimal Power Allocation Algorithm with Fixed Task Scheduling (OPA-FTS).

B. Task Scheduling for Fixed Power Allocation

With any given power allocation solution $\boldsymbol{P}$, $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ can be transformed into the following task scheduling problem:

$$\mathbf{P4}(\boldsymbol{x}): \ \max_{\boldsymbol{x}} \ \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{F}(j)}\left(\frac{1-\lambda}{W_{\max}}w_j - \frac{\lambda}{E_{\max}^{\mathrm{total}}}P_{s(j)}p_{jk}\right)x_{jk}^t \quad \text{s.t.} \ \ \mathrm{C3\text{-}C6}.$$

To solve $\mathbf{P4}(\boldsymbol{x})$ efficiently, we leverage a conflict graph model $G(\mathcal{V},\mathcal{E})$ with $\mathcal{V}$ and $\mathcal{E}$ respectively representing the vertex set and edge set, to depict the conflict relations among the decision variables.

We take the following two steps to construct our conflict graph $G(\mathcal{V},\mathcal{E})$. First, we construct vertex set $\mathcal{V}$ by representing any variable x_{jk}^t as a vertex v one by one. In this way, we establish one to one mapping between x_{jk}^t and vertex v. Furthermore, we assign a weight denoted by W_v for any vertex v. Specifically, we use the one to one mapping between vertex v and x_{jk}^t for any vertex v to find its corresponding task j and then assign $\frac{1-\lambda}{W_{\max}}w_j - \frac{\lambda}{E_{\max}^{\mathrm{total}}}P_{s(j)}p_{jk}$ as value of W_v. Second, from the constraints C3-C6 in $\mathbf{P4}(\boldsymbol{x})$, we can check whether any two variables x_{jk}^t (denoted by u) and $x_{j'k'}^{t'}$ (denoted by v) take the value of 1 at the same time. If not, we can add one edge across the vertexes u and v. We perform this procedure on all combinations on $\boldsymbol{x}$ to constitute the vertex set $\mathcal{E}$.

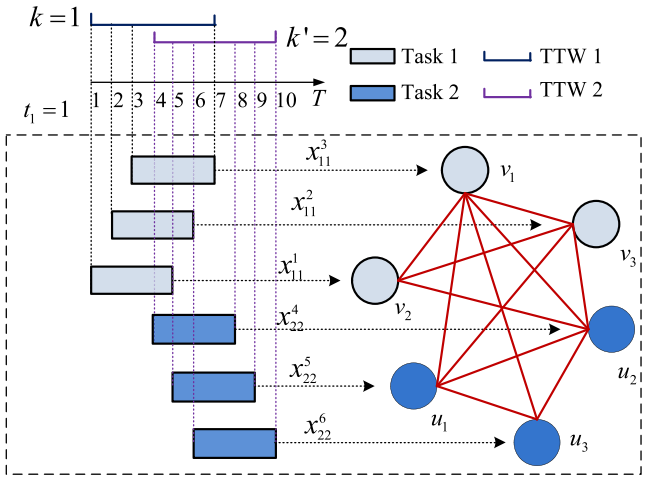


Fig. 3. The construction of conflict graph.

In Fig. 3, we illustrate the construction of conflict graph. From Fig. 3, we can observe that each task j within its associated time windows $\mathcal{K}_{s(j),\mathcal{H}}$ yields a set of vertexes, which constitutes a clique denoted by $\mathcal{C}_j$. For example, task 1 within time window 1 yields a vertex set of $\{v_1, v_2, v_3\}$. As such, we can obtain $\mathcal{C}_j = \{v_1, v_2, v_3\}$. Furthermore, we can see that two endpoints of any edge in $\mathcal{E}$ either belong to only one clique or two different cliques. As such, these cliques exactly divide the edge set $\mathcal{E}$ into two sets, denoted by $\mathcal{A}$ and $\mathcal{E}/\mathcal{A}$. In particular, $\mathcal{A}$ denotes the edge set, where any edge's endpoints are from two different cliques.

We now utilize the constructed conflict graph $G(\mathcal{V}, \mathcal{E})$ to rewrite $\mathbf{P4}(\boldsymbol{x})$ as a new form, which can then be efficiently solved via the SDR method. More precisely, we introduce decision variables $z_v \in \{0, 1\}$ to depict vertex assignment for $G(\mathcal{V}, \mathcal{E})$, so that $z_v = 1$ indicates that vertex $v \in \mathcal{V}$ is selected; and $z_v = 0$ otherwise. Then, $\mathbf{P4}(\boldsymbol{x})$ can be equivalently transformed into the following ILP problem:

$$\begin{aligned}
\mathbf{P5}(\boldsymbol{z}):\quad & \max_{\boldsymbol{z}} \sum_{v\in\mathcal{V}} W_v z_v \\
\text{s.t.}\quad & \text{C7}: \sum_{v\in\mathcal{C}_j} z_v \le 1,\ \forall j \in \mathcal{J}, \\
& \text{C8}: z_u + z_v \le 1,\ \forall (u,v) \in \mathcal{A}, \\
& \text{C9}: z_v \in \{0,1\},\ \forall v \in \mathcal{V},
\end{aligned}$$

where $\boldsymbol{z} = \{z_v\}$. In $\mathbf{P5}(\boldsymbol{z})$, C7 represents that at most a vertex is selected for any clique $\mathcal{C}_j$, C8 ensures that the two endpoints of any edge in $\mathcal{A}$ can not be selected at the same time, and C9 are binary constraints.

To obtain an SDR form, we first transform C9 into the following constraint:

$$\text{C10}: z_v(z_v - 1) = 0, \forall v \in \mathcal{V}.$$

Then, $\mathbf{P5}(\boldsymbol{z})$ can be transformed into the following equivalent Quadratic Program (QP) problem:

$$\begin{aligned}
\mathbf{P6}(\boldsymbol{z}):\quad & \max_{\boldsymbol{z}} \sum_{v\in\mathcal{V}} W_v z_v \\
\text{s.t.}\quad & \text{C7-C8, C10}.
\end{aligned}$$

We now transform $\mathbf{P6}(\boldsymbol{z})$ into a vector form to facilitate our analysis. We define $\boldsymbol{\eta} = [z_1, z_2, \ldots, z_{|\mathcal{V}|}]^T$ as a $1 \times |\mathcal{V}|$ column vector. Let $\boldsymbol{W} = [W_1, W_2, \ldots, W_{|\mathcal{V}|}]^T$. We denote $\boldsymbol{e}^v_{1\times|\mathcal{V}|}$ by a $|\mathcal{V}|$-dimensional column unit vector with the v-th element being 1. Moreover, we let $\boldsymbol{a}^{\mathcal{C}_j}_{1\times|\mathcal{V}|}$ denote a $|\mathcal{V}|$-dimensional column bit vector, whose each element is either 0 or 1. In particular, the v-th element of $\boldsymbol{a}^{\mathcal{C}_j}_{1\times|\mathcal{V}|}$ is set to 1 for any $v \in \mathcal{C}_j$; and 0 otherwise. We further define $\boldsymbol{b}_{u,v}$ as a $|\mathcal{V}|$-dimensional column bit vector with the u-th and the v-th elements being 1 and the remaining elements being 0. Using the above defined vectors, we can recast $\mathbf{P6}(\boldsymbol{z})$ as the following vector form:

$$\begin{aligned}
\mathbf{P7}(\boldsymbol{\eta}):\quad & \max_{\boldsymbol{\eta}}\ (\boldsymbol{W})^T\boldsymbol{\eta} \\
\text{s.t.}\quad & \left(\boldsymbol{a}^{\mathcal{C}_j}_{1\times|\mathcal{V}|}\right)^T\boldsymbol{\eta} \le 1,\ \forall j \in \mathcal{J}, \\
& \left(\boldsymbol{b}_{u,v}\right)^T\boldsymbol{\eta} \le 1,\ \forall (u,v) \in \mathcal{A}, \\
& (\boldsymbol{\eta})^T \operatorname{diag}\left(\boldsymbol{e}^v_{1\times|\mathcal{V}|}\right)\boldsymbol{\eta} = \left(\boldsymbol{e}^v_{1\times|\mathcal{V}|}\right)^T\boldsymbol{\eta},\ \forall v \in \mathcal{V}, \\
& \boldsymbol{\eta} \ge 0.
\end{aligned}$$

We further define $\boldsymbol{q} = [(\boldsymbol{\eta})^T, 1]^T$ to obtain the Quadratically Constrained Quadratic Program (QCQP) form of $\mathbf{P7}(\boldsymbol{\eta})$ as follows:

$$\begin{aligned}
\mathbf{P8}(\boldsymbol{q}):\quad & \max_{\boldsymbol{q}} \boldsymbol{q}^T\boldsymbol{G}^{\text{obj}}\boldsymbol{q} \\
\text{s.t.}\quad & \boldsymbol{q}^T\boldsymbol{G}^{\mathcal{J}}_j\boldsymbol{q} \le 1,\ \forall j \in \mathcal{J}, \\
& \boldsymbol{q}^T\boldsymbol{G}^{\mathcal{A}}_{u,v}\boldsymbol{q} \le 1,\ \forall (u,v) \in \mathcal{A}, \\
& \boldsymbol{q}^T\boldsymbol{G}^{\mathcal{V}}_v\boldsymbol{q} = 0,\ \forall v \in \mathcal{V}, \\
& \boldsymbol{q} \ge 0,
\end{aligned}$$

where

$$\boldsymbol{G}^{\text{obj}} = \begin{bmatrix} \boldsymbol{0} & \frac{1}{2}\boldsymbol{W} \\ \frac{1}{2}(\boldsymbol{W})^T & 0 \end{bmatrix},\ \boldsymbol{G}^{\mathcal{A}}_{u,v} = \begin{bmatrix} \boldsymbol{0} & \frac{1}{2}\boldsymbol{b}_{u,v} \\ \frac{1}{2}(\boldsymbol{b}_{u,v})^T & 0 \end{bmatrix},$$

$$\boldsymbol{G}^{\mathcal{J}}_j = \begin{bmatrix} \boldsymbol{0} & \frac{1}{2}\boldsymbol{a}^{\mathcal{C}_j}_{1\times|\mathcal{V}|} \\ \frac{1}{2}\left(\boldsymbol{a}^{\mathcal{C}_j}_{1\times|\mathcal{V}|}\right)^T & 0 \end{bmatrix},$$

$$\boldsymbol{G}^{\mathcal{V}}_v = \begin{bmatrix} \operatorname{diag}\left(\boldsymbol{e}^v_{1\times|\mathcal{V}|}\right) & -\frac{1}{2}\boldsymbol{e}^v_{1\times|\mathcal{V}|} \\ -\frac{1}{2}\left(\boldsymbol{e}^v_{1\times|\mathcal{V}|}\right)^T & 0 \end{bmatrix}.$$

Next, we define $\boldsymbol{Z} = \boldsymbol{q}\boldsymbol{q}^T$ to rewrite $\mathbf{P8}(\boldsymbol{q})$ into a Semi-Definite Program (SDP) form as follows:

$$\begin{aligned}
\mathbf{P9}(\boldsymbol{Z}):\quad & \max_{\boldsymbol{Z}}\ \operatorname{Tr}\left(\boldsymbol{G}^{\text{obj}}\boldsymbol{Z}\right) \\
\text{s.t.}\quad & \operatorname{Tr}\left(\boldsymbol{G}^{\mathcal{J}}_j\boldsymbol{Z}\right) \le 1,\ \forall j \in \mathcal{J}, \\
& \operatorname{Tr}\left(\boldsymbol{G}^{\mathcal{A}}_{u,v}\boldsymbol{Z}\right) \le 1,\ \forall (u,v) \in \mathcal{A}, \\
& \operatorname{Tr}\left(\boldsymbol{G}^{\mathcal{V}}_v\boldsymbol{Z}\right) = 0,\ \forall v \in \mathcal{V}, \\
& \boldsymbol{Z}(|\mathcal{V}|+1, |\mathcal{V}|+1) = 1, \\
& \boldsymbol{Z} \succeq 0, \\
& \operatorname{Rank}(\boldsymbol{Z}) = 1.
\end{aligned}$$

Although $\mathbf{P9}(\boldsymbol{Z})$ is still a non-convex problem that is difficult to solve, we can drop the rank constraint $\operatorname{Rank}(\boldsymbol{Z}) =$

Algorithm 2 SDR-Based Task Scheduling Algorithm With Fixed Power Allocation (SDR-TSFPA)

Initialization: $\tilde{\boldsymbol{\pi}}_n = \mathbf{0}$.
Drop the constraint of $\text{Rank}(\boldsymbol{Z}) = 1$ from $\mathbf{P9}(\boldsymbol{Z})$ and solve it to obtain an optimal solution $\boldsymbol{Z}^*$.
for $n = 1{:}N$ **do**
  Generate $\boldsymbol{\pi}_n \sim \mathcal{N}(0, \boldsymbol{Z}^*)$.
end for
Sort $\boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_N$ in decreasing order to obtain a new sequence of $\boldsymbol{\pi}'_1, ..., \boldsymbol{\pi}'_N$.
for $n = 1{:}N$ **do**
  if $\boldsymbol{\pi}'_n = 1$ satisfies the constraints of C4, C5, and C6 **then**
    $\tilde{\boldsymbol{\pi}}_n = \boldsymbol{\pi}'_n$.
  else
    $\tilde{\boldsymbol{\pi}}_n = 0$.
  end if
end for
Obtain n^* by solving $\max_n (\boldsymbol{W})^T \tilde{\boldsymbol{\pi}}_n$.
Output: $\tilde{\boldsymbol{\pi}}_{n^*}$.

1 to obtain a standard SDR form. Then, we can use some standard SDP solver to solve $\mathbf{P9}(\boldsymbol{Z})$ to obtain a solution $\boldsymbol{Z}^*$. If $\boldsymbol{Z}^*$ is an integer solution, then we obtain an optimal solution to $\mathbf{P9}(\boldsymbol{Z})$; otherwise, we need to convert it into an approximate integer solution to $\mathbf{P9}(\boldsymbol{Z})$ via a randomization method.

Inspired by [38], we use the Gaussian randomization method to yield a set of high quality approximate solutions. We assume the Gaussian randomization are performed N rounds. For any round $n \in \{1, 2, \ldots, N\}$, we use a normal distribution with zero expectation and variance $\boldsymbol{Z}^*$ to generate the n-th approximate solution, denoted by $\boldsymbol{\pi}_n$, i.e., $\boldsymbol{\pi}_n \sim \mathcal{N}(0, \boldsymbol{Z}^*)$. However, the generated approximate solution $\boldsymbol{\pi}_n$ is not always feasible for $\mathbf{P9}(\boldsymbol{Z})$. It is therefore necessary to map $\boldsymbol{\pi}_n$ into a feasible solution of $\mathbf{P9}(\boldsymbol{Z})$, denoted by $\tilde{\boldsymbol{\pi}}_n$. Let $\tilde{\boldsymbol{\pi}} = \{\tilde{\boldsymbol{\pi}}_n\}$. The elements of $\boldsymbol{\pi}$ are sorted in decreasing order. Starting from the largest one to the smallest one, we set each element $\boldsymbol{\pi}_n$ in $\boldsymbol{\pi}$ to 1 when it satisfies the constraints of C4, C5, and C6; and 0 otherwise. Then, we set $\tilde{\boldsymbol{\pi}_n} = \boldsymbol{\pi}_n, \forall n$. Finally, we obtain $\tilde{\boldsymbol{\pi}}_{n^*}$ corresponding to the largest objective value of $\mathbf{P9}(\boldsymbol{Z})$ among the N rounds of randomization. We summarize the above solution in Algorithm 2, termed as SDR-based Task Scheduling Algorithm with Fixed Power Allocation (SDR-TSFPA). Furthermore, the above procedure of problem transformation from **P0** to **P9** is illustrated in Fig. 4.

Remark 3: We note that $\mathbf{P4}(\boldsymbol{x})$ resembles a classic parallel machine scheduling with time windows. Consequently, this kind of problems can be converted into a special linear integer structure identical to $\mathbf{P5}(\boldsymbol{z})$ using the conflict graph model. This distinctive structure can be further resolved in an efficient manner using SDR.

C. Joint Power Allocation and Task Scheduling

In this section, we first use the result in Section IV-A to reduce the solution space of $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ without losing optimality and then introduce our proposed two-layer solution

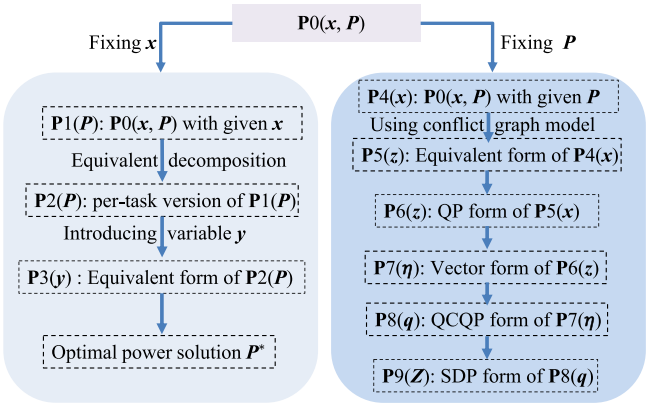


Fig. 4. Relationship diagram of problems.

based on genetic framework. We have shown that $\mathbf{P1}(\boldsymbol{P})$ can be solved optimally in a closed form (i.e., $\boldsymbol{P}^*$). Therefore, we can obtain the minimum total energy consumption for $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P}^*)$. Furthermore, we substitute (13) into (5) to obtain:

$$p_{jk} = \frac{D_j}{B_c \log_2\left(1 + \frac{P_{s(j)} G^{\text{tran}}_{s(j)} G^{\text{rec}}_h L_f L^k_l}{N_0}\right)}, \quad \forall j \in \mathcal{J}, k \in \mathcal{K}_{s(j),\mathcal{H}}, \tag{21}$$

which is inversely proportional to $P_{s(j)}$. This indicates that increasing the value $P_{s(j)}$ would shorten the transmitting time p_{jk}, further increasing the sum weights of tasks. As such, we can obtain that the optimal power allocation solution to $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ is in the smaller set of feasible power, i.e., $\mathcal{P} = \{P_{s(j)} | P^*_{s(j)} \le P_{s(j)} \le P^{\max}_{s(j)}, \ \forall j \in \mathcal{J}\}$.

To tackle $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ efficiently, we decompose it into two levels of optimization as follows. In the upper-level optimization, we adopt a genetic framework to solve $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ by optimizing variables $\boldsymbol{x}$ and $\boldsymbol{P}$. In the lower-level optimization, we use the proposed solution in Section IV-B to calculate the objective value of $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ with given variables $\boldsymbol{x}$ and $\boldsymbol{P}$. In order to use the genetic framework to remodel $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$, we require the following two definitions.

Genetic Representation of the Solutions to $\boldsymbol{P0}(\boldsymbol{x}, \boldsymbol{P})$: For convenient genetic representation, we first discretize the set of feasible power $\mathcal{P}$ to obtain a new set $\mathcal{P}_{\text{Discret}} = \{\mathcal{P}_{s(j)}\}$ with $\mathcal{P}_{s(j)} = \{P^m_{s(j)}\}$, each is calculated by

$$P^m_{s(j)} = P^*_{s(j)} + m \frac{P^{\max}_{s(j)} - P^*_{s(j)}}{\left|\mathcal{P}_{s(j)}\right|}, \tag{22}$$

where $m = 0, 1, \ldots, |\mathcal{P}_{s(j)}| - 1$. We can then represent each candidate discrete solution to $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ as a chromosome. Specifically, we observe from the definition of $\boldsymbol{x}$ that each chromosome is actually a three-dimensional binary vector. Thus, we adopt binary coding to execute chromosome coding, so that one-to-one mapping relationship is established with $(\boldsymbol{x}, \boldsymbol{P})$ and a chromosome. This is achieved by assigning 0 or 1 to each element of each chromosome to indicate task scheduling and power allocation.

Algorithm 3 Population Initialization (PInit) Algorithm

Input: Population size U, $\mathcal{U}=\emptyset$, w_j, $\mathcal{P}_{\text{Discret}}$.

1: Calculate the value of $\text{Prob}(\boldsymbol{x})=\{p(x_{jk}^t)\}$.
2: Obtain the set of data offloading tasks J_s for each LEO satellite s.
3: **for** $u=1{:}U$ **do**
4: **for** $s=1{:}S$ **do**
5: **for** $j_{id}=1{:}|J_s|$ **do**
6: Use the roulette wheel selection method with the probability of $p(x_{jk}^t)$ to get the scheduling order for task $J_s(j_{id})$ on LEO satellite s.
7: **end for**
8: Allocate feasible time slot $t\in\mathcal{F}(j)$ for each task j in the scheduling order of each LEO satellite s.
9: **end for**
10: Use steps 4-9 to obtain the value of $\boldsymbol{x}$.
11: Randomly generate $\boldsymbol{P}\in\mathcal{P}_{\text{Discret}}$.
12: Map $(\boldsymbol{x},\boldsymbol{P})$ into chromosome $\mathcal{U}_u$ and call SDR-TSFPA (i.e., Algorithm 2) to calculate its fitness value.
13: Update $\mathcal{U}=\mathcal{U}\cup\mathcal{U}_u$.
14: **end for**

Output: $\mathcal{U}$.

Algorithm 4 Energy-Efficient Data Offloading (EDO) Algorithm

Input: $P_{s(j)}^{\max}$, generation number L, population size U.

1: Obtain $\boldsymbol{P}^*=\{P_{s(j)}^*\}$ by calling OPA-FTS.
2: Utilize $\{P_{s(j)}^{\max}\}$ and $\boldsymbol{P}^*$ to obtain $\mathcal{P}_{\text{Discret}}$.
3: Call PInit to obtain an initial population $\mathcal{U}$.
4: **while** $l\le L$ **do**
5: Replicate a new population $\mathcal{U}'$.
6: Execute crossover and mutation operators on $\mathcal{U}'$.
7: Map each chromosome in $\mathcal{U}'$ into $(\boldsymbol{x},\boldsymbol{P})$ and then compute their fitness values by calling SDR-TSFPA.
8: Merge the two populations of $\mathcal{U}$ and $\mathcal{U}'$.
9: Use the strategies of tournament and elitism to select U chromosomes from $\mathcal{U}\cap\mathcal{U}'$.
10: $l=l+1$.
11: **end while**
12: Choose the best chromosome and then map it into $(\boldsymbol{x},\boldsymbol{P})$.

Output: $(\boldsymbol{x},\boldsymbol{P})$.

Quality Evaluation of the Represented Solutions: We define a fitness function over the genetic representation as the objective value in $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ to evaluate each candidate solution. The fitness of each chromosome can be computed as follows: First, the value of $\boldsymbol{P}$ for each chromosome is randomly generated in $\mathcal{P}_{\text{Discret}}$. Second, we obtain $\mathbf{P4}(\boldsymbol{x})$ by substituting the obtained $\boldsymbol{P}$ into $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$. Third, we can obtain the value of $\boldsymbol{x}$ from each chromosome through their one-to-one mapping relationship. Fourth, we use the value of $\boldsymbol{x}$ to construct a conflict graph $G(V,E)$ through the proposed method in Section IV-B. In particular, we represent each element of $\boldsymbol{x}$ equal to one as one vertex in $G(V,E)$. Then, we use SDR-TSFPA (i.e., Algorithm 2) to determine the value of $\boldsymbol{x}$. Finally, we substitute both the obtained values of $\boldsymbol{x}$ and $\boldsymbol{P}$ into the objective function in $\mathbf{P4}(\boldsymbol{x})$ to compute the fitness of each chromosome. Once the genetic representation and the fitness function are defined, we begin with an initial population of chromosomes and execute the repetitive application of the bio-inspired operations of mutation, crossover, and selection to evolve better solutions to $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$.

To solve $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ efficiently, we further propose a novel Population Initialization method (PInit), summarized in Algorithm 3. In detail, we first devise a population generation probability denoted by $\text{Prob}(\boldsymbol{x})=\{p(x_{jk}^t)\}$ to achieve the diversity of an initial population. We use $p(x_{jk}^t)$ to represent the generation probability for any variable x_{jk}^t, given by

$$p\left(x_{jk}^t\right)=\frac{w\left(x_{jk}^t\right)}{\sum_{j\in\mathcal{J}_s}w\left(x_{jk}^t\right)},\tag{23}$$

where $w(x_{jk}^t)$ indicates the mapping between variable x_{jk}^t and w_j, i.e., $w(x_{jk}^t)=w_j$ and $\mathcal{J}_s$ is the set of data offloading tasks of LEO satellite s. Second, we use $\text{Prob}(\boldsymbol{x})=\{p(x_{jk}^t)\}$ to generate a high-quality initial population. Specifically, we use $\text{Prob}(\boldsymbol{x})=\{p(x_{jk}^t)\}$ to sort the tasks in $\mathcal{J}_s$ for each LEO satellite s to determine the task scheduling order. Third, we allocate a feasible time slot in $\mathcal{F}(j)$ for each task j in the scheduling order of each LEO satellite s. Finally, we compute the fitness of the generated initial population via SDR-TSFPA (i.e., Algorithm 2).

In Algorithm 4, we summarize the above two-layer solution, termed as EDO. The complexity of EDO is dominated by calling SDR-TSFPA to calculate the fitness value for each chromosome in Step 7. The complexity of SDR-TSFPA is $\mathcal{O}((JT\sum_{s\in\mathcal{S}}|\mathcal{K}_{s,\mathcal{H}}|)^{4.5}\log(\frac{1}{\epsilon}))$, where ϵ is the solution accuracy [38]. The complexity of Step 7 is $\mathcal{O}(U(JT\sum_{s\in\mathcal{S}}|\mathcal{K}_{s,\mathcal{H}}|)^{4.5}\log(\frac{1}{\epsilon}))$. Thus, the total complexity of EDO is $\mathcal{O}(UL(JT\sum_{s\in\mathcal{S}}|\mathcal{K}_{s,\mathcal{H}}|)^{4.5}\log(\frac{1}{\epsilon}))$. Finally, we give the following theorem to reveal the convergence of EDO.

Theorem 2: The EDO algorithm is guaranteed to converge within a finite number of generations.

Proof: The EDO algorithm is a dual-layer optimization framework, incorporating an outer-layer GA and an inner-layer SDR-TSFPA. Within the inner-layer of the SDR-TSFPA, the interior-point algorithm is readily applicable to determine an optimal solution for the SDR version of $\mathbf{P9}(\boldsymbol{Z})$, achieving a linear convergence rate of $\mathcal{O}(\log(1/\epsilon))$ without any additional modifications [39]. This indicates that the convergence rate of the inner-layer SDR-TSFPA is linear, precisely at a rate of $\mathcal{O}(\log(1/\epsilon))$. Furthermore, the minorization condition within Markov chain theory serves as a key tool in [40] to analyze the convergence of GA. The analysis presented in [40] demonstrates that the outer-layer GA also exhibits linear convergence with a rate of $\mathcal{O}(\log(1/\epsilon))$. In light of the aforementioned analysis, it can be concluded that the EDO algorithm converges in a finite number of generations. ■

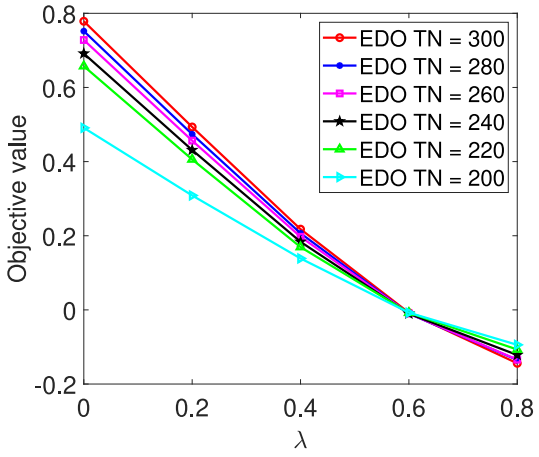


Fig. 5. Objective value versus λ.

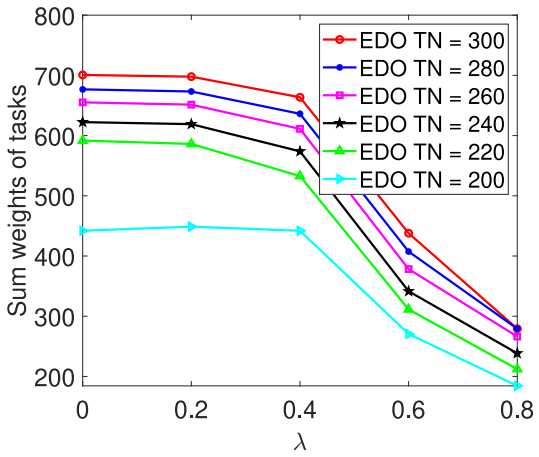


Fig. 6. Sum weights of tasks versus λ.

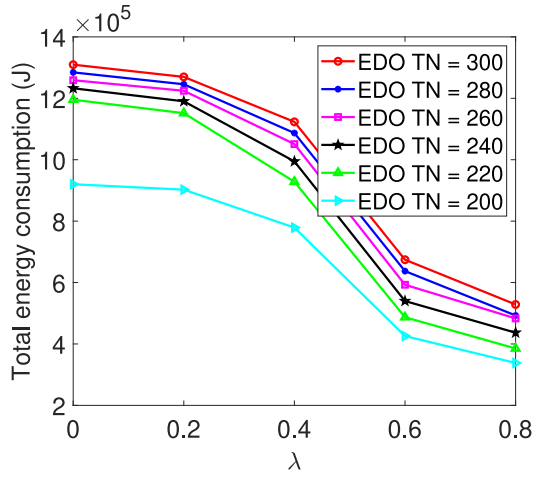


Fig. 7. Total energy consumption versus λ.

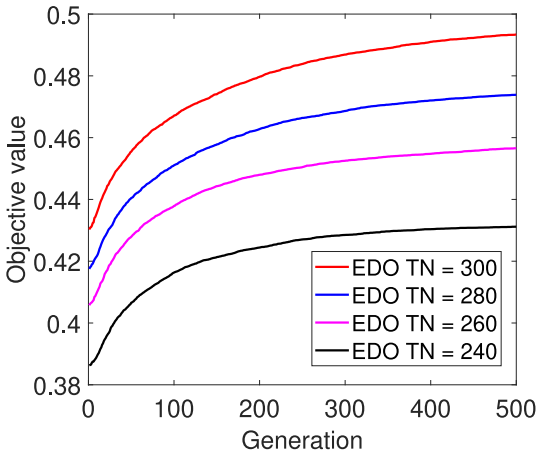


Fig. 8. Objective value versus generations.

V. Simulation Results

The simulation scenario consists of 3 GSs and 3 LEO satellites. We use the softwares of Satellite Tool Kit (STK) and MATLAB to form a co-simulation platform to produce the considered simulation scenario. The whole scheduling horizon is from 1 December 2024 00:00:00.0 to 1 December 2024 12:00:00.0. By assigning the numbers of North American Air Defense Command (NORAD), we use MATLAB to input the two-line elements of each LEO satellites into STK directly. The NORAD numbers of the LEO satellites are set to 31,113, 32,382, and 33,320. The locations of the GSs are set to (18° N, 109.5° E), (40° N, 116° E), and (39.5° N, 76° E). Each GS or LEO satellite is equipped with one antenna. The size of time slot is set to 10 seconds. The weight of each task j (i.e., w_j) is randomly distributed within the interval of $[1, 5]$. We set $G_{s(j)}^{\text{tran}} = G_h^{\text{rec}} = 36$ dB, $L_l^k = 1$, $L_f = 10^{-23}$, $N_0 = 1.2418 * 10^{-14}$ J, $B_c = 2.2$ GHz, $R_k^{\min} = 750$ Mbps, $P_{s(j)}^{\max} = 100$ Watts, $E_{\max}^{\text{total}} = 2.0 * 10^6$ J, and $W = 900$. The probabilities of crossover and mutation are set to 1 and 0.2, respectively. We set $U = 60$, $L = 500$, and $N = 10^4$. We use the solver of SDPNAL [41] to solve the SDR problem.

A. Scenario for the Uniform Distribution of Task Data

In this scenario, the data size of each task j (i.e., D_j) is uniformly generated from the interval of $[60, 180]$ Gbits. We adopt four experiments to verify the performance of the proposed EDO algorithm. First, we show the design objective value versus λ in Fig. 5, and then vary the values of λ to show the balance between two conflicting sub-objectives in Figs. 6 and 7. Second, we set $\lambda = 0.2$ and $\lambda = 0$ to evaluate the convergence of EDO in Figs. 8 and 9, respectively. Third, we set $\lambda = 0$ to study the impact of $P_{s(j)}^{\max}$ on the sum weights of tasks in Fig. 10. Finally, we set $\lambda = 0.2$ to evaluate the superiority of EDO in Figs. 11-13.

As depicted in Fig. 5, we show how the objective value varies with λ for different Task Numbers (TNs). From the objective value in Fig. 5, we further plot its two terms of the sum weights of tasks and the total energy consumption in Fig. 6 and Fig. 7, respectively. We can see that the objective value, the sum weights of tasks, and the total energy consumption decrease as λ grows. This result coincides well with our design objective function in $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$. This is because that a larger value of λ favors the objective of minimizing the total energy consumption, while a smaller one favors the objective of maximizing the sum weights of tasks. This reveals a balance

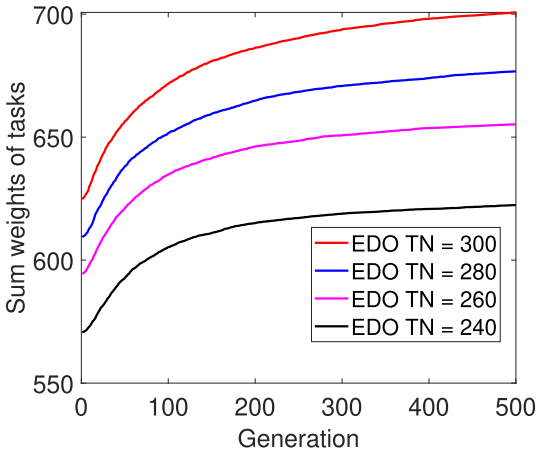


Fig. 9. Sum weights of tasks versus generations.

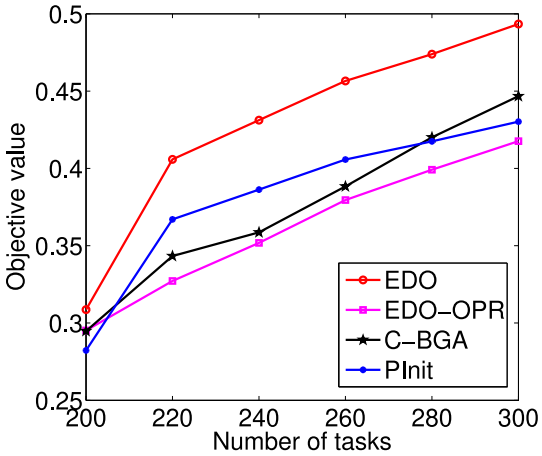


Fig. 11. Objective value versus the number of tasks.

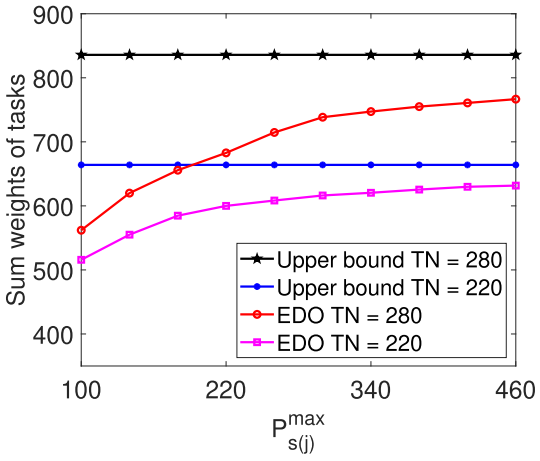


Fig. 10. Sum weights of tasks versus $P^{\max}_{s(j)}$.

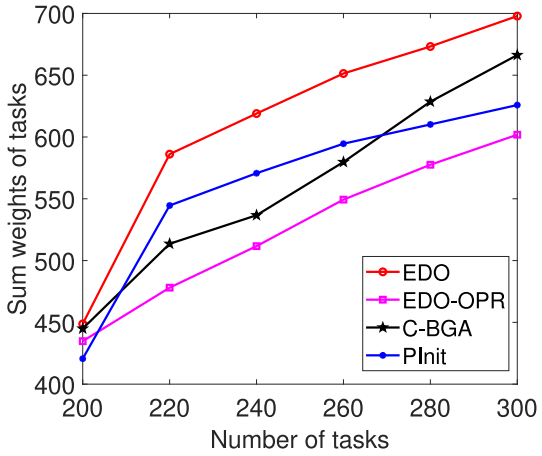


Fig. 12. Sum weights of tasks versus the number of tasks.

between the objectives of maximizing the sum weights of tasks and minimizing the total energy consumption. Furthermore, this phenomenon indicates that we can adjust the value of λ in practical applications to produce the desired metrics involving the sum weights of tasks and the total energy consumption.

In Fig. 8, we show the objective value versus generations for the different number of tasks. We further set $\lambda = 0$ in EDO to obtain the sum weights of tasks and plot it versus generations in Fig. 9. From Figs. 8 and 9, we observe that both the objective value and the sum weights of tasks first increase as the number of tasks increases and then remain constant after 400 generations. This demonstrates that the proposed EDO algorithm achieves fast convergence in limited generations.

Fig. 10 compares EDO with *Upper bound* for the different number of tasks in terms of the sum weights of tasks. The *Upper bound* can be obtained by calculating the sum weight of arrival tasks. We see that the sum weights of tasks first monotonically increases and then approaches to a constant value as $P^{\max}_{s(j)}$ grows. This is because that more large power is allocated for data offloading to reduce the transmission time of tasks as $P^{\max}_{s(j)}$ increases. The sum weights of tasks with increasing $P^{\max}_{s(j)}$ does not change because of the limited time window constraints. Furthermore, the performance gap of EDO and *Upper bound* gradually decreases as $P^{\max}_{s(j)}$ increases. This phenomenon indicates that properly increasing $P^{\max}_{s(j)}$ can improve the performance of NGOSNs.

We adopt the baselines of EDO with Original Power Range (EDO-OPR), C-BGA [27], and PInit to evaluate the performance of EDO. In EDO-OPR, we use EDO to solve the studied problem subject to the power range of $[0, P^{\max}_{s(j)}], \forall j$. In Fig. 11, we compare EDO with EDO-OPR, C-BGA, and PInit in terms of the objective value versus the number of tasks. It is observed that the proposed EDO algorithm substantially outperforms the comparison schemes of EDO-OPR and C-BGA in terms of the objective value as the number of tasks increase. This is because our proposed EDO algorithm jointly optimizes power allocation and task scheduling over a smaller feasible set of power obtained by OPA-FTS to maximize the designed objective of $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$. In addition, EDO achieves larger objective values than PInit with the increasing number of tasks. It is because that the better solutions can be evolved via the bio-inspired operations in EDO.

Furthermore, we plot Figs. 12 and 13 to show the two sub-objectives of the sum weights of tasks and the total energy consumption as the number of tasks increase, respectively. From Fig. 12, we observe that EDO substantially outperforms

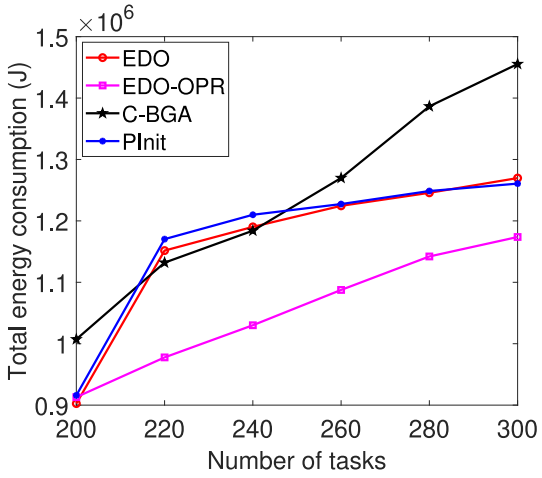


Fig. 13. Total energy consumption versus the number of tasks.

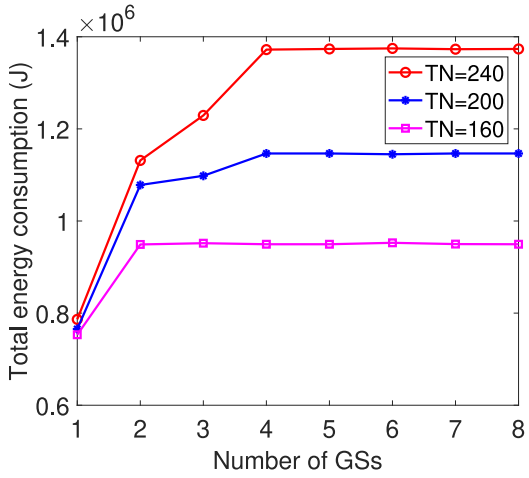


Fig. 15. Total energy consumption versus the number of GSs.

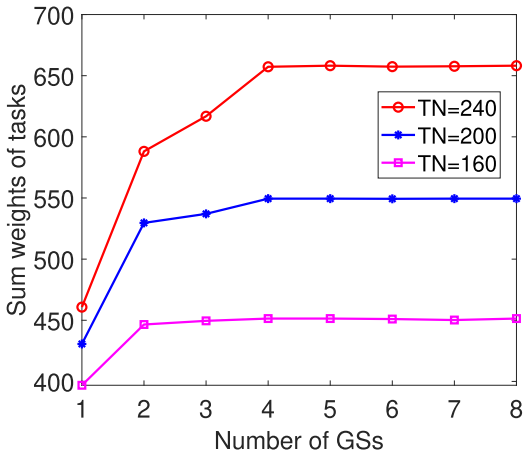


Fig. 14. Sum weights of tasks versus the number of GSs.

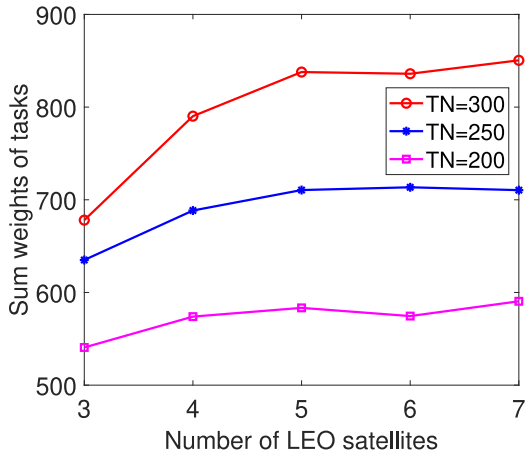


Fig. 16. Sum weights of tasks versus the number of LEO satellites.

the baselines of EDO-OPR, C-BGA, and PInit in terms of the sum weights of tasks, further demonstrating the superiority of our solution. In Fig. 13, EDO needs more total energy consumption than EDO-OPR with the number of task increasing. This is because that more successfully offloaded tasks need to consume more energy. In particular, a smaller $\lambda = 0.2$ in this experiment tends to maximize the sum weights of tasks instead of minimizing the total energy consumption. This indicates that a smaller λ is set to obtain more sum weights of tasks when the energy in NGOSNs is sufficient.

From Figs. 12 and 13, we see that compared with PInit, EDO is capable of maximizing the sum weights of tasks while reducing the total energy consumption. This further demonstrates the superiority of the bio-inspired operations used in EDO. As shown in Figs. 11-13, our proposed EDO can achieve the larger objective value and the larger sum weights of tasks over C-BGA in a lower total energy consumption, when the number of tasks is larger than 260. This is because the solution space of $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ enlarges as the number of tasks increase. As such, EDO can find some high-quality solutions in a larger solution space. This indicates that EDO is suitable for the NGOSNs with the large number of tasks.

B. Scenario for the Gaussian Distribution of Task Data

In this context, the data size for each task is randomly produced following a normal distribution. This distribution exhibits a mean μ of 120 Gbits and a variance σ^2 of 0.05μ. The values of $G_{s(j)}^{\text{tran}}$ and G_h^{rec} are randomly generated by an uniform function within the interval of [35, 37] dB. We conduct three experiments to assess the impact of various system parameters on the performance of the proposed EDO algorithm. In the first experiment, we evaluate the performance of the EDO algorithm in relation to the number of GSs, as depicted in Figs. 14 and 15. The coordinates of the eight GSs are set to (39.5° N, 76° E), (40° N, 116° E), (18° N, 109.5° E), (33.8° N, −106.7° E), (−23.2° N, −46.6° E), (41.3° N, −75.9° E), (36.8° N, −116.1° E), and (−3.9° N, −38.4° E). These GSs are successively input into STK to form a varying number of GSs. In the second experiment, we verify the performance of the EDO algorithm with respect to the number of LEO satellites, as shown in Figs. 16 and 17. The NORAD numbers of seven LEO satellites are set to 32,382, 31,113, 33,320, 28,254, 39,150, 40,053, and 25,994. The NORAD numbers from the first three to the first seven are used in sequence and input into STK to form varying numbers of LEO satellites. In the third experiment, we evaluate the

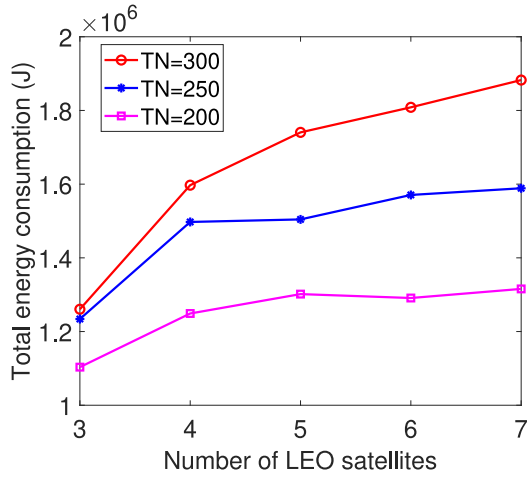


Fig. 17. Total energy consumption versus the number of LEO satellites.

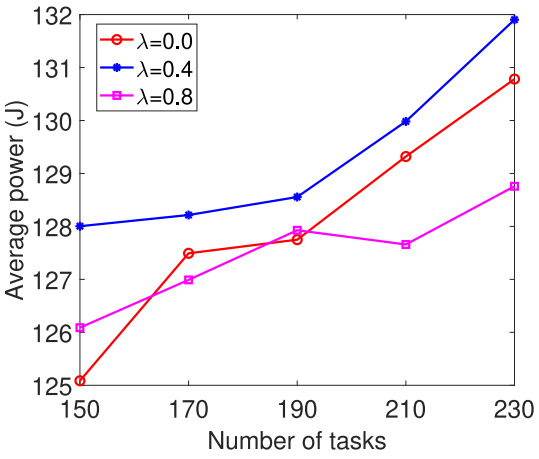


Fig. 18. Average power versus the number of tasks.

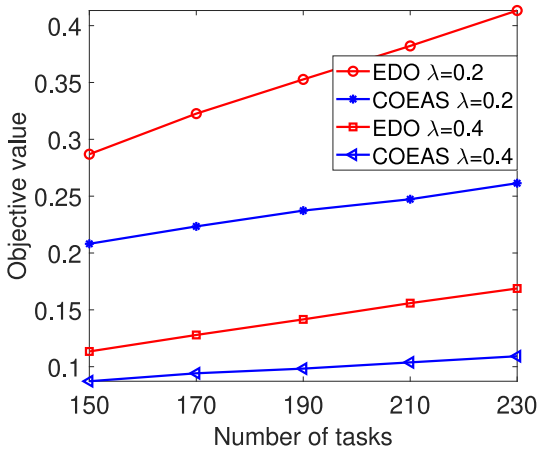


Fig. 19. Performance comparison in terms of Objective value.

performance of the EDO algorithm in relation to the number of tasks, as illustrated in Figs. 18 and 19.

Figs. 14 and 15 show the variations in the two sub-objectives of the sum weights of tasks and the total energy consumption, respectively, as the number of GSs increases. From Figs. 14 and 15, we observe that both the sum weights of tasks and the total energy consumption initially increase and eventually stabilize for the different number of tasks with the increased number of GSs. In particular, the rate of convergence is inversely proportional to the number of tasks. Specifically, the EDO algorithm achieves convergence at a GS number of two when the task number is 160, and at a GS number of four when the task number reaches 200. These observations suggest that the number of GSs should be adjusted in response to changes in the task quantity within real-world applications.

In Figs. 16 and 17, we plot the changes in the two sub-objectives of the sum weights of tasks and the total energy consumption as the number of LEO satellites increasing, respectively. As illustrated in Figs. 16 and 17, it is evident that the EDO algorithm can achieve a greater sum of task weights and higher total energy consumption as the number of LEO satellites increases. Nevertheless, this trend becomes less pronounced when the number of LEO satellites exceeds four. This is because growth in the number of LEO satellites increases the number of feasible TTWs, when the number of LEO satellites is relatively small. However, the number of feasible TTWs is predominantly constrained by the number of GSs, when the number of LEO satellites is comparatively substantial.

As shown in Fig. 18, we represent the relationship between the average power consumption of the EDO algorithm and the number of tasks for various λ. From Fig. 18, it is evident that the average power consumption of the EDO algorithm directly correlates with the increase in task number; it rises in conjunction with the increase in tasks. This finding suggests that the EDO algorithm, under conditions of increased network load, augments power allocation for tasks to facilitate the scheduling of a higher quantity of them.

We employ a CO-evolutionary algorithm with Elite Archive Strategy (COEAS) [28] to tackle $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$, chosen for its universal applicability. We compare the results of EDO and COEAS in terms of the relationship between the objective value and the number of tasks for different λ in Fig. 19. As illustrated in Fig. 19, it is evident that the proposed EDO algorithm significantly outperforms COEAS with respect to objective value, particularly as the number of tasks increases for different values of λ. This is attributed to the ability of SDR-TSFPA in the proposed EDO algorithm to secure a higher fitness value for each chromosome.

VI. Conclusion

We study energy-efficient data offloading in NGOSNs by jointly optimizing power allocation and task scheduling, to simultaneously minimize the total energy consumption and maximize the sum task weight. We systematically explore the unique structures of the joint optimization problem. We first consider two special cases for the joint optimization problem and derive their near-optimal solutions. We then combine the two special solutions in the genetic framework to propose an efficient two-layer energy-efficient data offloading algorithm. The simulation results demonstrate the feasibility and convergence of the proposed algorithm. The simulation results also show that the proposed algorithm can properly balance the two metrics of the total energy consumption and the sum weights of tasks and outperform the current literature.

REFERENCES

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

[2] (Cisco, San Francisco, CA, USA). *Cisco Edge-to-Enterprise IoT Analytics for Electric Utilities Solution Overview*. (2023). [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/big-data/solutionoverview-c22-740248.html

[3] "Study on new radio (NR) to support non-terrestrial networks; Version 15.4.0," 3GPP, Sophia Antipolis, France, Rep. TR 38.811, 2020.

[4] D. Zhou, M. Sheng, J. Li, and Z. Han, "Aerospace integrated networks innovation for empowering 6G: A survey and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 975–1019, 2nd Quart., 2023.

[5] M. Sheng et al., "Coverage enhancement for 6G satellite-terrestrial integrated networks: Performance metrics, constellation configuration and resource allocation," *Sci. China Inf. Sci.*, vol. 66, no. 3, pp. 1–20, Feb. 2023.

[6] J. Liang et al., "Latency versus transmission power trade-off in free-space optical (FSO) satellite networks with multiple inter-continental connections," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 3014–3029, 2023.

[7] L. He, J. Li, Y. Wang, J. Zheng, and L. He, "Balancing total energy consumption and mean makespan in data offloading for space-air-ground integrated networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 209–222, Jan. 2024.

[8] L. He, Z. Jia, K. Guo, H. Gan, Z. Han, and C. Yuen, "Online joint data offloading and power control for space-air-ground integrated networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 12, pp. 18126–18141, Dec. 2024.

[9] M. Minardi, T. X. Vu, L. Lei, C. Politis, and S. Chatzinotas, "Virtual network embedding for NGSO systems: Algorithmic solution and SDN-testbed validation," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 3523–3535, Sep. 2023.

[10] 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.

[11] S. D. Reddy and W. L. Brown, *Single Processor Scheduling with Job Priorities and Arbitrary Ready and Due Times*, Comput. Sci. Corp., Tysons, VA, USA, 1986.

[12] S. Rojanasoonthon, J. F. Bard, and S. D. Reddy, "Algorithms for parallel machine scheduling: A case study of the tracking and data relay satellite system," *J. Oper. Res. Soc.*, vol. 54, no. 8, pp. 806–821, Dec. 2017.

[13] S. Rojanasoonthon and J. Bard, "A GRASP for parallel machine scheduling with time windows," *INFORMS J. Comput.*, vol. 17, no. 1, pp. 32–51, Feb. 2005.

[14] B. Deng, C. Jiang, J. Wang, and L. Kuang, "Beam scheduling with various mission demands in data relay satellite systems," *J. Commun. Inf. Netw.*, vol. 6, no. 4, pp. 396–410, Dec. 2021.

[15] M. He, Y. Zhu, and X. Jia, "Scheduling model and heuristic algorithm for tracking and data relay satellite considering multiple slide windows," *J. Zhengzhou Univ.*, vol. 39, no. 5, pp. 11–21, Aug. 2018.

[16] C. Guo, W. Xiong, and L. Hao, "Relay satellite system task scheduling algorithm based on double-layer priority," *Appl. Res. Comput.*, vol. 35, no. 5, pp. 1506–1510, Jan. 2018.

[17] X. Chen, X. Li, X. Wang, Q. Luo, and G. Wu, "Task scheduling method for data relay satellite network considering break point transmission," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 844–875, Jan. 2021.

[18] G. Wu, Q. Lou, Y. Zhu, X. Chen, Y. Feng, and W. Pedrycz, "Flexible task scheduling in data relay satellite networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 2, pp. 1055–1068, Apr. 2022.

[19] J. Li, G. Wu, T. Liao, M. Fan, X. Mao, and W. Pedrycz, "Task scheduling under a novel framework for data relay satellite network via deep reinforcement learning," *IEEE Trans. Veh. Tech.*, vol. 72, no. 5, pp. 6654–6668, May 2023.

[20] D. Karapetyan, S. M. Minic, K. T. Malladi, and A. P. Punnen, "Satellite downlink scheduling problem: A case study," *Omega*, vol. 53, no. 1, pp. 115–123, Jun. 2015.

[21] Y. Song, L. Xing, M. Wang, Y. Yi, W. Xiang, and Z. Zhang, "A knowledge-based evolutionary algorithm for relay satellite system mission scheduling problem," *Comput. Ind. Eng.*, vol. 150, no. 1, pp. 1–11, Dec. 2020.

[22] N. Zufferey and M. Vasquez, "A generalized consistent neighborhood search for satellite range scheduling problems," *RAIRO-Oper. Res.*, vol. 49, no. 1, pp. 99–121, Jan. 2015.

[23] Z. Liu, Z. Feng, and Z. Ren, "Route-reduction-based dynamic programming for large-scale satellite range scheduling problem," *Eng. Optim.*, vol. 51, no. 11, pp. 1944–1964, Jan. 2019.

[24] Y. Chen, D. Zhang, M. Zhou, and H. Zou, "Multi-satellite observation scheduling algorithm based on hybrid genetic particle swarm optimization," in *Proc. AITIA*, 2012, pp. 441–448.

[25] Z. Zhang, F. Hu, and Z. Na, "Ant colony algorithm for satellite control resource scheduling problem," *Appl. Intell.*, vol. 48, no. 1, pp. 3295–3305, Feb. 2018.

[26] J. Zhang and L. Xing, "An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem," *Comput. Oper. Res.*, vol. 139, no. 1, pp. 1–11, Mar. 2022.

[27] Y. Song, J. Ou, J. Wu, Y. Wu, L. Xing, and Y. Chen, "A cluster-based genetic optimization method for satellite range scheduling system," *Swarm Evol. Comput.*, vol. 79, no. 1, pp. 1–12, Jun. 2023.

[28] M. Xiong, W. Xiong, and Z. Liu, "A co-evolutionary algorithm with elite archive strategy for generating diverse high-quality satellite range schedules," *Complex Int. Syst.*, vol. 9, no. 5, pp. 5157–5172, Mar. 2023.

[29] J. Qu et al., "Deep reinforcement learning method for satellite range scheduling problem," *Swarm Evol. Comput.*, vol. 77, no. 1, pp. 1–10, Mar. 2023.

[30] J. Liang et al., "A fast approach to satellite range rescheduling using deep reinforcement learning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 6, pp. 9390–9403, Dec. 2023.

[31] K. An, T. Liang, X. Yan, Y. Li, and X. Qiao, "Power allocation in land mobile satellite systems: An energy-efficient perspective," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1374–1377, Jul. 2018.

[32] Y. Ruan, Y. Li, C.-X. Wang, R. Zhang, and H. Zhang, "Energy efficient power allocation for delay constrained cognitive satellite terrestrial networks under interference constraints," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4957–4969, Oct. 2019.

[33] Z. Ji, S. Wu, C. Jiang, D. Hu, and W. Wang, "Energy-efficient data offloading for multi-cell satellite-terrestrial networks," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2265–2269, Oct. 2020.

[34] S. Zhang, G. Cui, and W. Wang, "Joint data downloading and resource management for small satellite cluster networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 887–901, Jan. 2022.

[35] A. Golkar and I. L. i. Cruz, "The federated satellite systems paradigm: Concept and business case evaluation," *Acta Astron.*, vol. 111, pp. 230–248, Jun/Jul. 2015.

[36] W.-C. Lin, D.-Y. Liao, C.-Y. Liu, and Y.-Y. Lee, "Daily imaging scheduling of an earth observation satellite," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 2, pp. 213–223, Mar. 2005.

[37] C. Lin and B.-S. Chen, "Achieving Pareto optimal power tracking control for interference limited wireless systems via multi-objective H_2/H_∞ optimization," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6154–6165, Dec. 2013.

[38] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

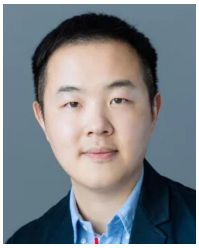
[39] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM J. Optim.*, vol. 6, no. 2, pp. 342–361, May 1996.

[40] J. He and L. Kang, "On the convergence rate of genetic algorithms," *Theor. Comput. Sci.*, vol. 229, nos. 1–2, pp. 23–39, Nov. 1999.

[41] D. Sun, K.-C. Toh, Y. Yuan, and X.-Y. Zhao, "SDPNAL+: A MATLAB software for semidefinite programming with bound constraints (version 1.0)," 2019, *arXiv:1710.10604*.

Lijun He (Member, IEEE) received the B.S. degree in electronic information science and technology from Anqing Normal University, Anhui, China, in 2013, and the Ph.D. degree in military communications from the State Key Laboratory of ISN, Xidian University, Xi'an, China, in 2020. From September 2018 to September 2019, he was with the University of Toronto, Toronto, ON, Canada, as a Visiting Scholar funded by the China Scholarship Council. From June 2020 to July 2022, he was a Postdoctoral Researcher with the School of Software, Northwestern Polytechnical University, where he was an Associate Professor from July 2022 to September 2024. He is currently an Associate Professor with the School of Information and Control Engineering, China University of Mining and Technology. His current research interests include routing, scheduling, resource allocation, and satellite communications.

Ziye Jia (Member, IEEE) received the B.E., M.S., and Ph.D. degrees in communication and information systems from Xidian University, Xi'an, China, in 2012, 2015, and 2021, respectively. From 2018 to 2020, she was a Visiting Ph.D. Student with the Department of Electrical and Computer Engineering, University of Houston. She is currently an Associate Professor with the Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her current research interests include space-air-ground networks, aerial access networks, resource optimization, and machine learning.



Juncheng Wang (Member, IEEE) received the B.Eng. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014, the M.Sc. degree in electrical and computer engineering from the University of Alberta,Edmonton, AB, Canada, in 2017, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2023. He is currently an Assistant Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, China. His research interests include network artificial intelligence, distributed machine learning, and online optimization.

Erick Lansard received the Engineering master's degree in aerospace, the Master of Science degree in fluid mechanics, and the Ph.D. degree in space geodesy in 1983, 1983, and 1987, respectively. In 2010, he was appointed as a VP R&T France and the Director of Thales Research & Technology-France, the Thales Corporate Research Lab, Palaiseau. In October 2013, he was appointed as a VP Technical & Space Development with Thales Solutions Asia, Singapore (also in charge of Innovation). From 2017 to February 2022, he was a VP Innovation and R&T of Thales Defense Mission Systems (also in charge of open innovation collaborations worldwide). Since March 2022, he has been a Full Professor with Nanyang Technological University with the mission to develop NTU Space Activities. His research interests include space systems & technologies, space sciences, and space applications.

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

From 2000 to 2002, he was a R&D Engineer with JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Idaho. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department, Computer Science Department, University of Houston, Texas. 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. He received the 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, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, the IEEE Vehicular Technology Society 2022 Best Land Transportation Paper Award, and the several best paper awards in IEEE conferences. 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. He is a 1% highly cited researcher since 2017 according to Web of Science. He has been an AAAS Fellow since 2019, and an ACM Fellow since 2024.

Chau Yuen (Fellow, IEEE) received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2000 and 2004, respectively. He was a Postdoctoral Fellow with Lucent Technologies Bell Labs, Murray Hill, in 2005. From 2006 to 2010, he was with the Institute for Infocomm Research, Singapore. From 2010 to 2023, he was with the Engineering Product Development Pillar, Singapore University of Technology and Design. Since 2023, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, where he is currently a Provost's Chair in wireless communications, and an Assistant Dean with Graduate College. He has four U.S. Patents and published over 400 research papers at international journals. He received the IEEE Communications Society Leonard G. Abraham Prize in 2024, the IEEE Communications Society Best Tutorial Paper Award in 2024, the IEEE Communications Society Fred W. Ellersick Prize in 2023, the IEEE Marconi Prize Paper Award in Wireless Communications in 2021, the IEEE APB Outstanding Paper Award in 2023, and the EURASIP Best Paper Award for Journal on Wireless Communications and Networking in 2021. He currently serves as an Editor-in-Chief for *Springer Nature Computer Science*, an Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE SYSTEM JOURNAL, and the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, where he was awarded as an IEEE TNSE Excellent Editor Award and the Top Associate Editor for TVT from 2009 to 2015. He also served as a Guest Editor for several special issues, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS MAGAZINE, IEEE COMMUNICATIONS MAGAZINE, IEEE VEHICULAR TECHNOLOGY MAGAZINE, the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and *Elsevier Applied Energy*. He is a Distinguished Lecturer of IEEE Vehicular Technology Society, Top 2% Scientists by Stanford University, and also a Highly Cited Researcher by Clarivate Web of Science.