Energy-Efficient Data Offloading for Earth Observation Satellite Networks

Lijun He[*], Ziye Jia[†], Juncheng Wang[‡], Feng Wang[§], Erick Lansard[¶], and Chau Yuen [¶]
[*] School of Software, Northwestern Polytechnical University, China
[†] Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, China
[‡] Department of Computer Science, Hong Kong Baptist University, China
[§] Information Systems Technology and Design, Singapore University of Technology and Design, Singapore
[¶] School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
Email: lijunhe@nwpu.edu.cn, jiaziye@nuaa.edu.cn, jcwang@comp.hkbu.edu.hk, feng2_wang@sutd.edu.sg,
{chau.yuen, erick.lansard}@ntu.edu.sg

2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring) | 979-8-3503-8741-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-SPRING62846.2024.10682994

Abstract—**In Earth Observation Satellite Networks (EOSNs) with a large number of battery-carrying satellites, proper power allocation and task scheduling are crucial to improving the data offloading efficiency. As such, we jointly optimize power allocation and task scheduling to achieve energy-efficient data offloading in EOSNs, aiming to balance the objectives of reducing the total energy consumption and increasing the sum weights of tasks. First, we derive the optimal power allocation solution to the joint optimization problem when the task scheduling policy is given. Second, leveraging the conflict graph model, we transform the original joint optimization problem into a maximum weight independent set problem when the power allocation strategy is given. Finally, we utilize the genetic framework to combine the above special solutions as a two-layer solution for the joint optimization problem. Simulation results demonstrate that our proposed solution can properly balance the sum weights of tasks and the total energy consumption, thus achieving superior system performance over the current best alternatives.**

I. Introduction

Driven by the fast proliferation of remote sensing applications, such as environment monitoring, meteorology, and natural disaster surveillance, Earth Observation Satellite Networks (EOSNs) experience an explosive growth in remote sensing data [1]–[3]. There arises an urgent need to offload the collected remote sensing data from EOSNs to Ground Stations (GSs) for further data processing. Different from traditional ground wireless networks, EOSNs belong to Ad Hoc networks with dynamic network topologies [4]. This means that data offloading in EOSNs occurs only within limited Transmission Time Windows (TTWs) between Earth Observation Satellites (EOSs) and GSs due to the high-speed motion of EOSs. The key to efficient data offloading in EOSNs is how to schedule data offloading tasks within limited TTWs to maximize the desired network performance [5].

From the mathematical perspective, the data offloading problem in EOSNs is similar to the Parallel Machine Scheduling Problem with Time Windows (PMSPTW) [6]. Recently, a majority of works studied the PMSPTW to select a set of optimal GSs for Low Earth Orbit (LEO) satellites to schedule Satellite Ground Links (SGLs) while satisfying the time window constraints [7]–[13]. The scheduling algorithms proposed in these works are either deterministic [7], [8] or non-deterministic [9]–[13]. The deterministic algorithms use mathematical tools, such as tree search [7] and dynamic programming [8], to find an optimal solution under mild assumptions. However, these deterministic algorithms are time-consuming especially for large-scale problems and thus are not suitable for practical EOSNs. To accommodate large-scale EOSNs, some non-deterministic algorithms have been proposed to find approximation solutions with lower computation costs. They are mainly based on meta-heuristic methods, such as particle swarm optimization [9], ant colony optimization [10], and genetic algorithm [11]. Recently, some attempts have been made to utilize Deep Reinforcement Learning (DRL) for solving satellite scheduling problems. For example, the PMSPTW was transformed into an Assignment Problem (AP) and a Single Antenna Scheduling Problem (SASP) [12]. Then, the DRL method and a heuristic scheduling method were proposed to solve the AP and the SASP, respectively. In [13], a DRL-based algorithm was proposed for a rapid satellite range rescheduling problem to process real-time emergency events. However, the above works ignored the energy management of battery-carrying LEO satellites powered by the solar panels, resulting in data offloading interruptions due to the overuse of energy. As such, energy consumption control is crucial for efficient data offloading in EOSNs [14].

To improve data offloading efficiency, some researchers studied energy-efficient offloading for various satellite networks, such as land mobile satellite systems [15], cognitive satellite terrestrial networks [16], small satellite cluster networks [17], and satellite-terrestrial networks [18]. 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,

This work was supported in part by the National Natural Science Foundation of China under Grant 62201463 and 62301251, in part by Basic Research Programs of Taicang under Grant TC2021JC25, in part by the Natural Science Foundation of Jiangsu Province of China under Project BK20220883.

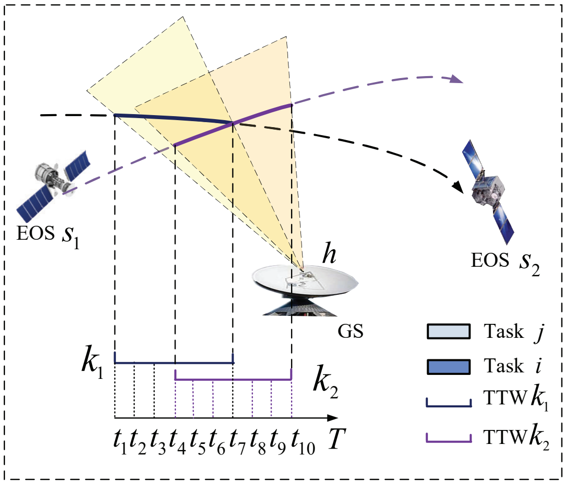


Fig. 1. Illustration of data offloading in EOSNs.

these works [15]–[18] are unsuitable for EOSNs to offload the tasks with a large amount of data.

In contrast to previous works [15]–[18], in this work, we character the main features of data offloading in real EOSNs 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 time window constraints of data offloading in EOSNs, which captures the intermittent characteristic of SGLs. Third, we develop an efficient yet fast two-layer optimization solution, termed Energy-efficient Data Offloading (EDO), to minimize the total energy consumption while maximizing the sum weights of tasks.

II. System Model and Problem Formulation

We consider a downlink EOSN consisting of a set $\mathcal{S} = \{1, ..., S\}$ of EOSs and a set of GSs. We discretize the whole scheduling horizon into T time slots to obtain a set of time slots, denoted by $\mathcal{T} = \{1, ..., T\}$. We denote by time slot t the time interval of $[t, t+1]$. We assume that each EOS and each GS is respectively equipped with one transmit antenna and one receiver antenna. Let $\mathcal{H} = \{1, ..., H\}$ denote the set of data receiver antennas. As shown in Fig. 1, EOSs cycle their individual orbits at high speed, thereby yielding a set of short contact TTWs. Let $\mathcal{K}_{s,h}$ denote the set of TTWs between each EOS $s \in \mathcal{S}$ and each 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, respectively. We denote by $\mathcal{J} = \{1, ..., J\}$ the set of data offloading tasks. Let $s(j)$ index the EOS that stores the data of task j.

A. Channel Model

We denote by R_{jk}^{SGL} the achievable data rate of SGLs within TTW k in bits/s for any task j, given by

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

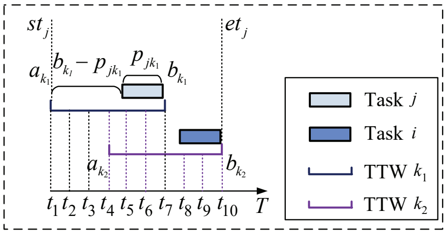


Fig. 2. Illustration of the set of feasible time slots within TTW k.

where B_c is the bandwidth and SNR_{jk} is the signal-to-noise ratio of offloading task j within TTW k. We evaluate the SNR_{jk} of SGLs [19] as follows:

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

where $G_{s(j)}^{\text{tran}}$ represents the transmit antenna gain of EOS $s(j)$, G_h^{rec} denotes the gain of receiver antenna h, L_f is the free space loss, N_0 is the noise power, and L_l^k is the total path loss for TTW k. The total path loss in dB is calculated by $L_l^k = L_b^k + L_g^k + L_s^k + L_e^k$, where L_b^k denotes the basic path loss in dB, L_g^k represents the attenuation due to atmospheric gasses in dB, L_s^k indicates the attenuation due to either ionospheric or tropospheric scintillation in dB, and L_e^k is the building entry loss in dB for TTW k [20]. In particular, $P_{s(j)}$ represents the transmit power of EOS $s(j)$ allocated to task j, which is subject to the following transmit power range of each EOS:

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

Furthermore, we let R_k^{req} represent the rate requirement of TTW k and introduce the following constraint to indicate the rate requirement of each TTW:

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

B. Task Model

Let D_j and w_j denote the data amount of task j and the weighted value of task j, respectively. We denote by p_{jk} the processing time, i.e., the transmitting time, of task j within TTW k, which can be computed by

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

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 the data of task j is offloaded in time slot t within TTW k; and $x_{jk}^t = 0$ otherwise. Thus, we introduce the following binary constraint

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

to indicate whether task j starts to offload in time slot t within TTW k, or not. Wherein, $\mathcal{F}(j)$ represents the set of all feasible time slots for task j, given by

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

with $\mathcal{F}(j,k) = [a_k, b_k - p_{jk}] \cap [st_j, et_j]$ indicating the set of feasible slots within TTW k to start offloading the data of task j. As an example for illustration in Fig. 2, we have $\mathcal{F}(j,k) = \{t_1, t_2, ..., t_5\}$.

We further introduce the following constraint to guarantee that each task j is offloaded at most once:

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

In addition, we denote by E_j the energy consumption for offloading task j, which is calculated by

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

C. Time Window Model

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

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

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

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

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

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

We further use the following constraint to indicate that any EOS offloads the data of at most one task in any time slot:

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

D. Problem Formulation

To balance the two conflicting objectives of minimizing the total energy consumption and maximizing the sum weights of tasks, we consider their normalized weighted sum and formulate the following mixed integer programming problem:

$$\begin{aligned} \mathbf{P0}(\boldsymbol{x},\boldsymbol{P})\text{:} \quad & \min_{\boldsymbol{x},\boldsymbol{P}} \ \frac{\lambda}{E_{\max}^{\text{total}}} \sum_{j\in\mathcal{J}} E_j - \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{F}(j)} \frac{1-\lambda}{W_{\max}} w_j x_{jk}^t \\ & \text{s.t.} \ \ \text{C1-C6}, \end{aligned} \tag{12}$$

where $\boldsymbol{x} = \{x_{jk}^t\}$, $\boldsymbol{P} = \{P_{s(j)}\}$, $W_{\max} = \sum_{j\in\mathcal{J}} w_j$ denotes the sum weights of all the tasks, $E_{\max}^{\text{total}}$ represents the sum of the maximization energy consumption of all tasks, and $\lambda \in [0,1)$ is a tuning weighted coefficient to obtain the desired total energy consumption and sum weights of tasks.

III. Proposed Solution

In this section, we first study two cases for $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ to obtain two special solutions and then utilize the genetic framework to combine them as an efficient two-layer solution.

A. Power Allocation for Fixed Task Scheduling

With a given task scheduling decision $\boldsymbol{x}$, $\mathbf{P0}(\boldsymbol{x},\boldsymbol{P})$ degrades into the total energy consumption minimization problem:

$$\begin{aligned} \mathbf{P1}(\boldsymbol{P}): \quad \eta(\boldsymbol{x}) \ = \ & \min_{\boldsymbol{P}} \ \frac{\lambda}{E_{\max}^{\text{total}}} \sum_{j\in\mathcal{J}} E_j - \mathcal{U} \\ & \text{s.t.} \ \ \text{C1-C2}, \end{aligned}$$

where $\mathcal{U}$ is a constant, given by

$$\mathcal{U} = \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{F}(j)} \frac{1-\lambda}{W_{\max}} w_j x_{jk}^t. \tag{13}$$

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

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

In line with (1) and (2), we obtain

$$R_{jk}^{\text{SGL}} = B_c \log_2\left(1 + \frac{P_{s(j)} G_{s(j)}^{\text{tran}} G_h^{\text{rec}} L_f L_l^k}{N_0}\right), \quad \forall j \in \mathcal{J}, k \in \mathcal{K}_{s(j),\mathcal{H}}. \tag{14}$$

Combining (5) and (9), we have

$$E_j = \sum_{t\in\mathcal{F}(j)} \frac{x_{jk}^t P_{s(j)} D_j}{R_{jk}^{\text{SGL}}}, \quad \forall j \in \mathcal{J}. \tag{15}$$

We combine (14) with (15) to obtain:

$$E_j = \frac{\alpha P_{s(j)}}{\log_2(1+\beta P_{s(j)})}, \quad \forall j \in \mathcal{J}, \tag{16}$$

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

$$y_j = \frac{1}{\log_2(1+\beta P_{s(j)})}, \quad \forall j \in \mathcal{J}. \tag{17}$$

We further use $\boldsymbol{y}$ to recast $\mathbf{P2}(\boldsymbol{P})$ as the following new form:

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

where $f(P_{s(j)}^{\max}) = \frac{1}{\log_2(1+\beta P_{s(j)}^{\max})}$ and $g(y_j) = y_j(2^{\frac{1}{y_j}} - 1)$. Next, we can obtain the first-order and second-order derivation of $g(y_j)$ as follows:

$$\nabla g(y_j) = 2^{\frac{1}{y_j}} \left(1 - \frac{\ln 2}{y_j}\right) - 1, \tag{18}$$

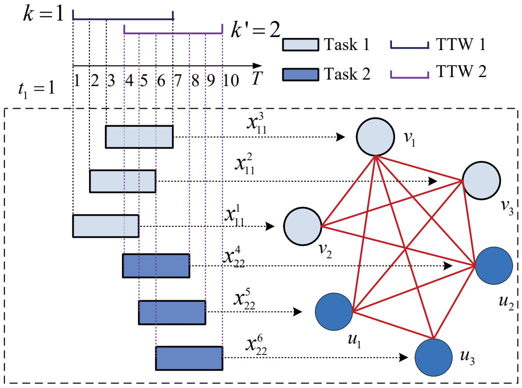


Fig. 3. The construction of conflict graph.

$$\nabla^2 g(y_j) = 2^{\frac{1}{y_j}} \frac{(\ln 2)^2}{(y_j)^3}. \tag{19}$$

For any y_j subject to $f(P^{\max}_{s(j)}) \leq y_j \leq \frac{B_c}{R^{\text{req}}_k}$, 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 $f(P^{\max}_{s(j)})$, $\frac{B_c}{R^{\text{req}}_k}$, or the extreme point of $g(y_j)$, depending on the value of $\hat{y}_j$, given by

$$y^*_j = \begin{cases} f(P^{\max}_{s(j)}) & \hat{y}_j < f(P^{\max}_{s(j)}), \\ \hat{y}_j & f(P^{\max}_{s(j)}) \leq \hat{y}_j < \frac{B_c}{R^{\text{req}}_k}, \\ \frac{B_c}{R^{\text{req}}_k} & \frac{B_c}{R^{\text{req}}_k} \leq \hat{y}_j, \end{cases} \tag{20}$$

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

$$\boldsymbol{P}^* = \{P^*_{s(j)} | P^*_{s(j)} = \frac{1}{\beta}(2^{y^*_j} - 1),\ \forall j \in \mathcal{J}\}. \tag{21}$$

B. Task Scheduling for Fixed Power Allocation

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

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

To solve $\mathbf{P4}(\boldsymbol{x})$ efficiently, we leverage a conflict graph model $G(V, E)$ with V and E respectively representing the vertex set and edge set, to depict the conflict relations among variables. We take the following two steps to construct conflict graph $G(V, E)$. First, we construct vertex set V by representing any variable x^t_{jk} as a vertex v one by one. As such, we establish one to one mapping between x^t_{jk} and v. Second, from the constraints of $\mathbf{P4}(\boldsymbol{x})$, we can check whether any two variables x^t_{jk} (denoted by u) and $x^{t'}_{j'k'}$ (denoted by v) take the value of one 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 vertex set E. We give an example for illustrating the conflict graph construction in Fig. 3. By constructing the conflict graph model $G(V, E)$, we can equivalently transform $\mathbf{P4}(\boldsymbol{x})$ into a Maximum Weight Independent Set (MWIS) problem .

Algorithm 1 Energy-efficient Data Offloading (EDO)

```
Input: $P^{\max}_{s(j)}$, generation number $L$, population size $U$.
Obtain $\boldsymbol{P}^*$ by calling the solution in Section III-A.
Utilize $\{P^{\max}_{s(j)}\}$ and $\boldsymbol{P}^*$ to obtain $\mathcal{P}_{\text{Discret}}$.
Use $\mathcal{P}_{\text{Discret}}$ to map $(\boldsymbol{x}, \boldsymbol{P})$ into a chromosome.
Adopt random scheme to obtain an initial population $\mathcal{U}$.
Compute the fitness of each chromosome in $\mathcal{U}$ by calling the solution in Section III-B.
while $l \leq L$ do
    Replicate a new population $\mathcal{U}'$.
    Execute crossover and mutation operators on $\mathcal{U}'$.
    Map each chromosome in $\mathcal{U}'$ into $(\boldsymbol{x}, \boldsymbol{P})$ and then compute their fitness values.
    Merge the two populations of $\mathcal{U}$ and $\mathcal{U}'$.
    Use the strategies of tournament and elitism to select $U$ chromosomes from $\mathcal{U} \cup \mathcal{U}'$.
    $l = l + 1$.
end while
Choose the best chromosome and then map it into $(\boldsymbol{x}, \boldsymbol{P})$.
Output: $(\boldsymbol{x}, \boldsymbol{P})$.
```

C. Joint Power Allocation and Task Scheduling

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

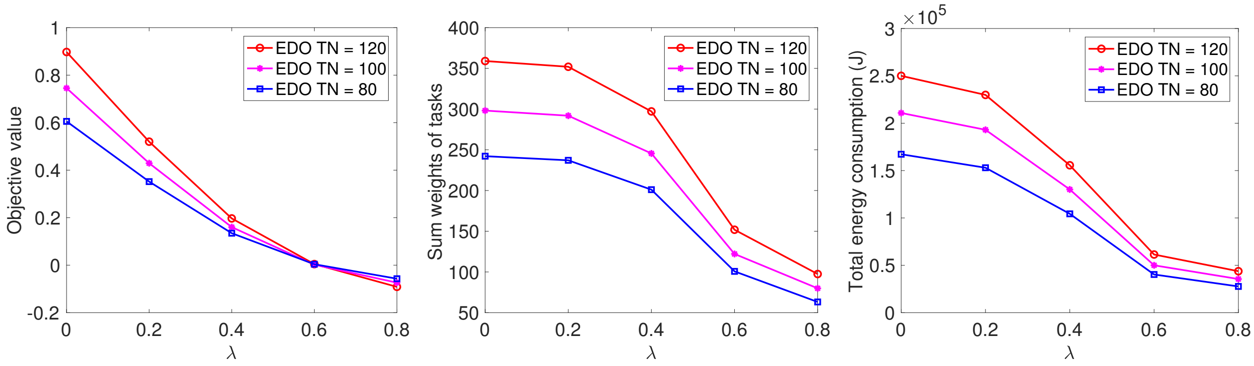
We have shown that $\mathbf{P1}(\boldsymbol{P})$ can be solved optimally in a closed form (i.e., $\boldsymbol{P}^*$). Therefore, we can obtain the minimum total energy consumption for $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P}^*)$. Furthermore, we substitute (14) into (5) to obtain:

$$p_{jk} = \frac{D_j}{B_c \log_2(1 + P_{s(j)}\beta)}, \quad \forall j \in \mathcal{J}, k \in \mathcal{K}_{s(j),\mathcal{H}}, \tag{22}$$

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 for $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ is in the smaller set of feasible power, i.e., $\mathcal{P} = \{P_{s(j)} | P^*_{s(j)} \leq P_{s(j)} \leq P^{\max}_{s(j)},\ \ \forall j \in \mathcal{J}\}$.

To tackle $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ efficiently, we decompose it into the 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 III-B to calculate the objective value of $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ with given variables $\boldsymbol{x}$ and $\boldsymbol{P}$. To 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 $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$: 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 $\mathcal{P}_{s(j)} = \{P^k_{s(j)} | P^k_{s(j)} = P^*_{s(j)} + k\frac{P^{\max}_{s(j)} - P^*_{s(j)}}{|\mathcal{P}_{s(j)}|}, k =$



(a) Objective value versus λ.

(b) Sum weights of tasks versus λ.

(c) Total energy consumption versus λ.

Fig. 4. Comparisons for different λ.

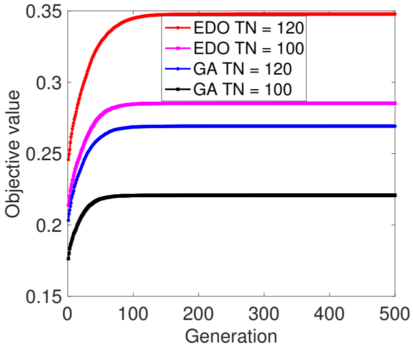


Fig. 5. Objective value versus generations.

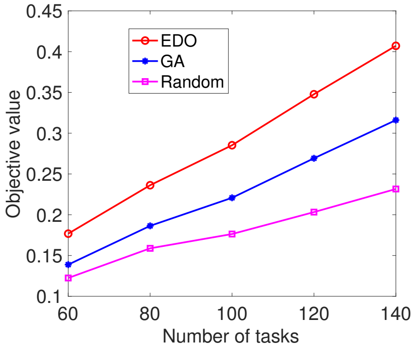


Fig. 6. Objective value versus the number of tasks.

$0, 1, ..., \left|\mathcal{P}_{s(j)}\right| - 1\}$. As such, we can 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, such that one-to-one mapping relationship is established with $(\boldsymbol{x}, \boldsymbol{P})$ and a chromosome, by assigning zero or one to each element of each chromosome to indicate task scheduling and power allocation.

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, we can obtain $\mathbf{P4}(\boldsymbol{x})$ by substituting known $\boldsymbol{P}$ into $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$. Second, we can obtain the value of $\boldsymbol{x}$ from each chromosome through their one-to-one mapping relationship. Third, we use the value of $\boldsymbol{x}$ to construct a conflict graph $G(V, E)$ through the proposed method in Section III-B. In particular, we represent each element of $\boldsymbol{x}$ equal to one as one vertex in $G(V, E)$ and compute a MWIS of $G(V, E)$. Then, we set the elements of $\boldsymbol{x}$ corresponding to the vertexes within MWIS to be ones; and zeros otherwise. Finally, we substitute both the obtained value 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, Genetic Algorithm (GA) can begin with an initial population of chromosomes and then evolves better solutions to $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$ through the repetitive application of the bio-inspired operations of mutation, crossover, and selection. We term the above two-layer solution as EDO, which is summarized in Algorithm 1.

IV. Simulation Results

The simulation scenario consists of 4 GSs and 4 EOSs. We use the softwares of satellite tool kit (STK) and MATLAB to form a co-simulation platform to produce the considered simulation scenario. Specifically, by assigning the numbers of north American air (NORAD), we use MATLAB to input the two-line elements of each EOS into STK directly. The NORAD numbers of the four EOSs are set to 31113, 32382, 33320, and 32289. The locations of the four GSs are set to $(18°\text{N}, 109.5°\text{E})$, $(40°\text{N}, 116°\text{E})$, $(34°\text{N}, 108°\text{E})$, and $(39.5°\text{N}, 76°\text{E})$. Each GS or EOS is equipped with one antenna. The whole scheduling horizon is set to 12 hours. The size of time slot is set to 10 seconds. The data size of each task j (i.e., D_j) is uniformly generated from the interval of $[5, 15]$ Gbits. The weight of each task j (i.e., w_j) is randomly distributed within the interval of $[1, 5]$. We set $G^{\text{tran}}_{s(j)} = G^{\text{rec}}_h = 36$ dB, $L^k_l = 1$, $L_f = 10^{-23}$, $N_0 = 5.16 * 10^{-20}$ J, $B_c = 2.2$ GHz, $R^{\text{req}}_k = 250$ Mbps, $P^{\max}_{s(j)} = 100$ Watts, $E^{\text{total}}_{\max} = 2.5 * 10^5$ J, and $W = 400$. The probabilities of crossover and mutation for GA are set to 0.5 and 0.8, respectively. We set $U = 60$ and $L = 200$.

We adopt two experiments to verify the performance of the proposed EDO algorithm. In the first experiment, we first show the design objective value versus λ in Fig. 4a, and then vary the values of λ to show the balance between two conflicting sub-objectives in Fig. 4b and Fig. 4c. In the second experiment, we set $\lambda = 0.3$ to respectively evaluate the convergence and superiority of EDO in Fig. 5 and Fig. 6. We adopt the baselines of *GA* and *Random* to evaluate the performance of EDO. In *GA*, we use the solution in Section III-B to calculate the fitness value for each chromosome and power allocation is subject to the interval of $[0, P^{\max}_{s(j)}], \forall j$. In *Random*, task scheduling and power allocation are randomly selected.

As depicted in Fig. 4a, we show how the objective value obtained by EDO varies with λ for different task number (TN). From the objective value in Fig. 4a, we further plot its two terms of the sum weights of tasks and the total energy consumption in Fig. 4b and Fig. 4c, 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 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. 5, we further show the comparisons between EDO and *GA* in terms of the objective value versus generations. From Fig. 5, we observe that the objective values obtained by both EDO and *GA* first increase with the number of tasks and then remain constant after 150 generations. This demonstrates that EDO achieves fast convergence in limited generations and provides a better convergence performance over *GA*.

In Fig. 6, we compare EDO with *GA* and *Random* in terms of the objective value versus the number of tasks. It is observed that EDO substantially outperforms the two comparison schemes in terms of the objective value as the number of tasks increase. This is because EDO jointly optimizes power allocation and task scheduling over a smaller feasible set of power to maximize the designed objective of $\mathbf{P0}(\boldsymbol{x}, \boldsymbol{P})$. From Fig. 6, we can see that the objective value monotonically increases with the number of tasks increase. This is because that a smaller λ tends to maximize the sum weights of tasks instead of minimizing the total energy consumption.

V. Conclusion

We study energy-efficient data offloading in EOSNs by jointly optimizing power allocation and task scheduling to minimize the total energy consumption and maximize the sum weights of tasks. We systematically explore the unique structures of joint optimization problem. We consider two special cases for the joint optimization problem and derive their near-optimal solutions. Then, we combine the two special solutions with genetic framework to propose an efficient two-layer EDO algorithm. Simulation results demonstrate the feasibility and effectiveness of the proposed algorithm.

References

[1] J. Du, C. Jiang, Q. Guo, M. Guizani, and Y. Ren, “Cooperative earth observation through complex space information networks,” *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 136–144, Apr. 2016.

[2] L. He, B. Liang, J. Li, and M. Sheng, “Joint observation and transmission scheduling in agile satellite networks,” *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4381–4396, Dec. 2022.

[3] 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, Feb. 2023.

[4] J. Li, Z. Haas, and M. Sheng, “Capacity evaluation of multi-channel multi-hop ad hoc networks,” in *IEEE ICPWC*, New Delhi, India, Dec. 2002, pp. 211–214.

[5] B. Deng, C. Jiang, L. Kuang, S. Guo, J. Lu, and S. Zhao, “Two-phase task scheduling in data relay satellite systems,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1782–1793, Feb. 2018.

[6] 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. Technol.*, vol. 72, no. 5, pp. 6654–6668, May 2023.

[7] 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.

[8] 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.

[9] Y. Chen, D. Zhang, M. Zhou, and H. Zou, “Multi-satellite observation scheduling algorithm based on hybrid genetic particle swarm optimization,” in *AITIA*, Berlin, Heidelberg, Jan. 2012, pp. 441–448.

[10] 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.

[11] 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.

[12] J. Qu, L. Xing, Y. Feng, M. Li, L. Jimin, Y. He, Y. Song, J. Wu, and G. Zhang, “Deep reinforcement learning method for satellite range scheduling problem,” *Swarm Evol. Comput.*, vol. 77, no. 1, pp. 1–10, Jan. 2023.

[13] J. Liang, J.-P. Liu, Q. Sun, Y.-H. Zhu, Y.-C. Zhang, J.-G. Song, and B.-Y. He, “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.

[14] D. Zhou, M. Sheng, Y. Zhu, J. Li, and Z. Han, “Mission QoS and satellite service lifetime tradeoff in remote sensing satellite networks,” *IEEE Wireless Commun. Lett.*, vol. 9, no. 7, pp. 990–994, Jul. 2020.

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

[16] 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.

[17] 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.

[18] X. Gao, J. Wang, X. Huang, Q. Leng, Z. Shao, and Y. Yang, “Energy-constrained online scheduling for satellite-terrestrial integrated networks,” *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2163–2176, Apr. 2023.

[19] A. Golkar and I. Lluch i Cruz, “The federated satellite systems paradigm: Concept and business case evaluation,” *Acta Astron.*, vol. 111, pp. 230–248, Jun. 2015.

[20] G. T. 38.811, “Study on new radio (NR) to support non-terrestrial networks,” Tech. Rep., Sept. 2020, V15.4.0.