This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3565571

Joint Data Compression and Task Scheduling for LEO Satellite Networks

Lijun He, *Member, IEEE*, Shiyin Li, *Member, IEEE*, Ziye Jia, *Member, IEEE*, Juncheng Wang, *Member, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—We investigate a joint data compression and task scheduling problem for Low Earth Orbit Satellite Networks (LEOSNs), to maximize the sum weights of tasks while simultaneously minimizing the total data loss. First, we propose a novel Multi-Resource Conflict Graph (MRCG) model to characterize the intertwined communication and computation allocation conflicts inherent in typical data offloading processes of LEOSNs. Leveraging the proposed MRCG model, we then formulate the studied problem for LEOSNs as a linear integer program, to maximize the normalized weighted sum of both the sum task weight and the total data loss. We further explore the intrinsic structure of the linear integer program to propose an efficient solution using the Semi-Definite Relaxation (SDR) technique. Finally, the simulation results underscore that the synergistic optimization of data compression and task scheduling significantly facilitates the data offloading efficiency of LEOSNs, and the performances surpass existing benchmarks across a broad range of system parameters.**

***Index Terms*—Low earth orbit satellite networks, task scheduling, data compression, semi-definite relaxation.**

I. Introduction

With the proliferation of global information, satellite networks have emerged as a crucial space-based component for 5G, and beyond 6G networks due to their global ubiquitous coverage ranges and high data transmission rates [1]–[3]. Compared to medium earth orbit and geostationary orbit satellites, Low Earth Orbit (LEO) satellites provide shorter communication delays, lower launch costs, and increased networking flexibility, largely attributable to their proximity to the ground. Due to these advantages, numerous commercial entities, such as SpaceX, Telesat, and Eutelsat, have launched a significant number of LEO satellites to establish LEO Satellite Networks (LEOSNs) [4]. Currently, LEOSNs are extensively employed in various fields such as wireless communication and earth observation, contributing to the rapid expansion of space tasks.

In contrast, the low-altitude orbits of LEO satellites in LEOSNs lead to high orbital velocity and limited footprint coverage. As a result, the contact time windows between LEO satellites and the data sinks of Ground Stations (GSs) and Data Relay Satellites (DRSs) are intermittent and restricted over time. This significantly reduces the data offloading efficiency of LEOSNs. Consequently, the primary challenge in enhancing the performance of LEOSNs is: how to efficiently allocate these intermittent and restricted contact time windows to accommodate the rapid increase in space tasks [5].

To circumvent this challenge, existing studies on the task scheduling for LEOSNs can be divided into two categories. The first category employs the time window model [6]–[8] or the graph model [9]–[11] to represent the intermittent communication links, simplifying the problem formulation and solution for LEOSN task scheduling. The second category focuses on maximizing the desired performance of LEOSNs, by fully utilizing limited communication link resources from various perspectives, such as multiple satellite collaboration [12] and data acquisition side [13]. However, all these works [6]–[13] only consider the optimization of communication resources for LEOSNs, neglecting the impact of computation resources on their optimization frameworks. This oversight often results in unresolved task transmission conflicts within time windows, thereby degrading the network performance of LEOSNs [14].

To this end, we seamlessly integrate lossy compression techniques with LEOSN task scheduling. This integration can effectively compress the data of tasks to shorten their transmission time and save communication resources. Furthermore, it can also mitigate task transmission conflicts within time windows. However, data compression may lead to data loss. Consequently, it is imperative to develop a novel scheduling strategy for LEOSNs that optimizes network performance while minimizing data loss. As such, we simultaneously optimize data compression and task scheduling within LEOSNs, with the objective of maximizing the combined weights of tasks and minimizing total data loss, all while adhering to the constraints of the time windows.

Our main contributions are summarized as follows.

- We propose a novel Multi-Resource Conflict Graph

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. (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, Shaanxi, 710071, China (e-mail: lijunhe@cumt.edu.cn).

Shiyin Li is with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China. (e-mail: lishiyin@cumt.edu.cn).

Ziye Jia is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210000, China (e-mail: jiaziye@nuaa.edu.cn).

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

Zhu Han is with the Department of Electrical and Computer Engineering at the 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).

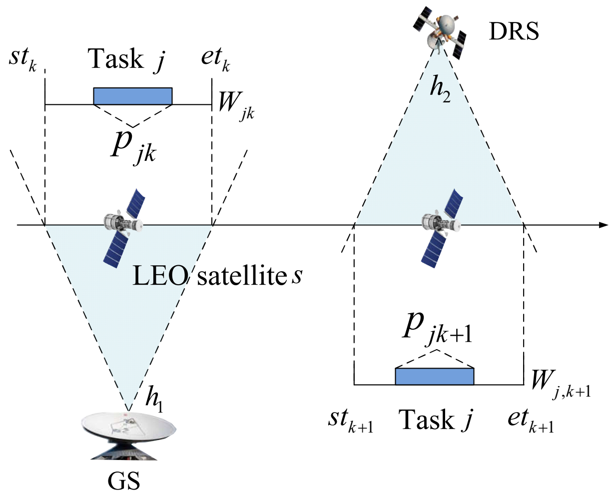

Fig. 1. Data offloading in LEOSNs.

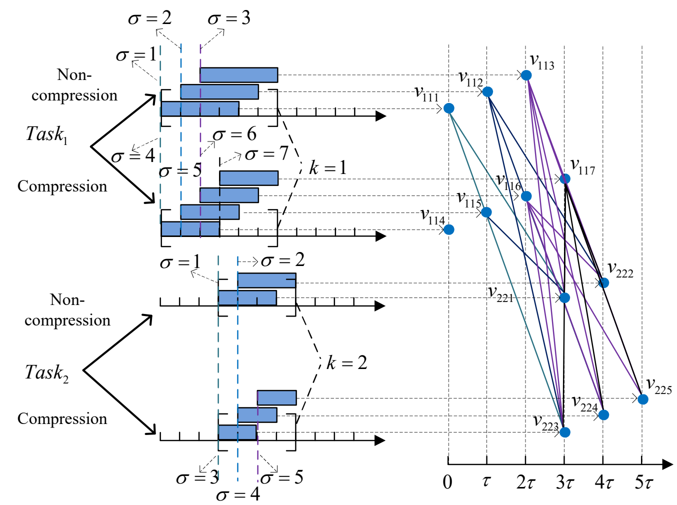

Fig. 2. An illustration of the proposed MRCG model.

(MRCG) model to delineate the task transmission conflicts during the joint allocation of communication and computation in LEOSNs.

- Based on the proposed MRCG model, we formulate the joint data compression and task scheduling problem for LEOSNs as a Linear Integer Program (LIP), to maximize the normalized weighted combination of the sum task weight and the total data loss.
- We explore the inherent structure of the formulated LIP, and propose an efficient solution grounded in Semi-Definite Relaxation (SDR).
- We conduct extensive simulation results to evaluate the necessity of jointly optimizing data compression and task scheduling in LEOSNs and the effectiveness of the proposed solution.

The rest of this work is structured as follows. Section II elaborates on the system model and the MRCG model. Our problem formulation is then presented in Section III. The proposed solution is introduced in Section IV. Section V provides simulation results to assess the performance of the proposed solution. Finally, concluding remarks are provided in Section VI.

II. System Model and MRCG Model

A. System Model

As shown in Fig. 1, we consider a LEOSN comprised of LEO satellites, DRSs, and GSs. The whole time axis is consistently segmented into a set $\mathcal{T}=\{1,2,...,T\}$ of time slots. Within this time-slotted network, a set $\mathcal{S}=\{1,2,...,S\}$ of LEO satellites offload a set $\mathcal{J}=\{1,2,...,J\}$ of the data offloading tasks to a set of data sinks including DRSs and GSs. We denote by $\mathcal{H}=\{1,2,...,H\}$ the set of data receiver antennas on the data sinks. The set $\mathcal{K}_{s(j),\mathcal{H}}$ represents the set of the time windows between LEO satellite $s(j)$ and $|\mathcal{H}|$ data receiver antennas, where $s(j)$ refers to the LEO satellite storing the data of task j. We use a 5-tuple $Task_j=(s(j),a_j,d_j,D_j,w_j)$ to represent task j. Here, a_j signifies the earliest start time for offloading task j, d_j specifies the latest end time for offloading task j, D_j refers to the data amount of task j, and w_j represents the weight value of task j. We define p_{jk} as the transmission duration required for task j to be offloaded within the time window $k\in\mathcal{K}_{s(j),\mathcal{H}}$, calculated by

$$p_{jk}=\frac{c_jD_j}{R_{jk}},\forall c_j\in\mathcal{C}_{s(j)},j\in\mathcal{J},k\in\mathcal{K}_{s(j),\mathcal{H}}, \tag{1}$$

where c_j is the compression ratio of task j, $\mathcal{C}_{s(j)}$ denotes the compression ratio set of LEO satellite s responsible for carrying the data of task j, and R_{jk} is the data offloading rate of task j within time window k.

As shown in Fig. 1, LEO satellite $s\in\mathcal{S}$ has the option to offload the data of task j either to a receiver antenna $h_1\in\mathcal{H}$ on a DRS via Inter-Satellite Links (ISLs) or to a receiver antenna $h_2\in\mathcal{H}$ on a GS via Satellite Ground Links (SGLs). The rapid movement of LEO satellites results in intermittent ISLs and SGLs. To characterize these two types of intermittent communication links, we introduce the time window model. In particular, we employ a 2-tuple (st_k,et_k) to represent the time window $k\in\mathcal{K}_{s(j),\mathcal{H}}$ between LEO satellite $s(j)$ and data receiver antenna h, where st_k and et_k signify the start and end times of time window k, respectively. We further designate $s(k)$ and $h(k)$ to index the LEO satellite s and data receiver antenna h associated with time window k. Based on these definitions, we use a 2-tuple $W_{jk}=(\alpha_{jk},\beta_{jk})$ to denote the feasible time window k for task j, where $\alpha_{jk}=\max(a_j,st_k)$ and $\beta_{jk}=\min(a_j,et_k)$.

B. MRCG Model

In LEOSNs, the process of data offloading with time windows typically encompasses both data compression and transmission. Upon the arrival of a task at a LEO satellite, it is necessary to determine the appropriate compression ratio and the optimal time for efficient data offloading. Consequently, computation and communication resources are intertwined and may conflict within specific time windows throughout the data offloading procedure in LEOSNs. To illustrate this point, we present an example in Fig. 2, which depicts the

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3565571

allocation of multiple resources within time windows during the data offloading process. Subsequently, we introduce a MRCG model to represent the resource conflict relationships in the data offloading process of LEOSNs, thereby facilitating problem formulation and resolution.

As shown in Fig. 2, two tasks necessitate scheduling within their specific time window, each of which can be compressed prior to offloading. From Fig. 2, we see that $Task_1$ can be started to be offloaded in four time slots when compressed, and three time slots when not compressed. This suggests that the introduction of computation resources can augment the flexibility of communication resource allocation, thereby improving the data offloading efficiency of LEOSNs. However, this multi-resource joint allocation also leads to a plethora of resource allocation conflicts. The accurate characterization of these resource conflict relationships is crucial to the data offloading strategy. In light of this, we propose the MRCG model, denoted by $G(\mathcal{V},\mathcal{E})$, to depict the evolving resource conflict relationships over time slots, where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the edge set. The construction of MRCG $G(\mathcal{V},\mathcal{E})$ involves two primary steps as follows.

In the first step, we construct the vertex set for any given task j denoted by $\mathcal{V}_j=\{v_{jk\sigma},\forall k\in\mathcal{K}_{s(j),\mathcal{H}}\}$. Subsequently, the overall vertex set is denoted as $\mathcal{V}=\{\mathcal{V}_j,\forall j\in\mathcal{J}\}$. Specifically, each vertex $v_{jk\sigma}=(\eta^{st}_{jk},\eta^{et}_{jk})\in\mathcal{V}_j$ is a 2-tuple to indicate that task j is scheduled within time window k with the sequencing number σ. Here, η^{st}_{jk} is the start time for offloading task j within time window k, while η^{et}_{jk} denotes the end time. This 2-tuple accurately characterizes the allocated communication and computation resources for any task j. More precisely, $v_{jk\sigma}=(\eta^{st}_{jk},\eta^{et}_{jk})$ represents the start and end times for allocating communication resources to task j. Furthermore, the transmission time p_{jk} can be derived using $(\eta^{st}_{jk},\eta^{et}_{jk})$ as $p_{jk}=\eta^{et}_{jk}-\eta^{st}_{jk}$, which, when integrated with (1), yields: $c_j=\frac{(\eta^{et}_{jk}-\eta^{st}_{jk})R_{jk}}{D_j}\in\mathcal{C}_{s(j)}$. This suggests that the compression ratio of task j can be derived from $v_{jk\sigma}=(\eta^{st}_{jk},\eta^{et}_{jk})$, which is essentially the computation resource allocation strategy. That is, any vertex $v_{jk\sigma}\in\mathcal{V}$ has the potential to characterize the multi-resource allocation for any task $j\in\mathcal{J}$.

To avoid confusion, we will omit the subscript of any vertex $v_{jk\sigma}$ henceforth, except in instances where it is necessary to determine the time window index, the task index, and the sequencing number. As such, we designate $k(v)$ and $j(v)$ to respectively index the time window and the data offloading task for vertex $v\in\mathcal{V}$. Furthermore, we employ $\eta^{st}_{jk}(v)$ and $\eta^{et}_{jk}(v)$ to denote the start time and the end time for any vertex v, respectively.

In the second step, an edge $e\in\mathcal{E}$ is represented as a 2-tuple (u,v), indicating that vertexes u and v are in conflict. In the practical data offloading of LEOSNs, two distinct types of task transmission conflicts are identified. The first type relates to the constraint that a single LEO satellite can only offload up to one task within a given time slot, represented by the edge set $\mathcal{E}_1$. The construction of the set $\mathcal{E}_1$ is as follows: If $s(u)=s(v)$ and $[\eta^{st}_{jk}(u),\eta^{et}_{jk}(u)]\cap[\eta^{st}_{jk}(v),\eta^{et}_{jk}(v)]\neq\emptyset$ for any pair of vertexes u and v, then $\mathcal{E}_1$ is updated to $\mathcal{E}_1=\mathcal{E}_1\cup(u,v)$. The second category stipulates that any data receiver antenna can only receive the data from a single task within any given time slot. This is represented by the edge set of $\mathcal{E}_2$. The construction of $\mathcal{E}_2$ is defined as follows: An edge is added to $\mathcal{E}_2$ if two conditions are satisfied for a pair of vertices, denoted as u and v. Firstly, it requires that the data receiver antennas associated with u and v are identical, i.e., $h(u)=h(v)$. Secondly, there must be a non-empty intersection between the time intervals associated with u and v, specifically, $[\eta^{st}_{jk}(u),\eta^{et}_{jk}(u)]\cap[\eta^{st}_{jk}(v),\eta^{et}_{jk}(v)]\neq\emptyset$. When these conditions are met, the edge set $\mathcal{E}_2$ is updated to include the pair (u,v), thereby becoming $\mathcal{E}_2\cup(u,v)$. The final edge set of the graph $G(\mathcal{V},\mathcal{E})$ is then derived as $\mathcal{E}=\mathcal{E}_1\cup\mathcal{E}_2$.

Remark 1. In $G(\mathcal{V},\mathcal{E})$, the vertex set $\mathcal{V}$ represents all possible allocations of communication and computation resources, while the edge set $\mathcal{E}$ denotes conflicts in these resource allocations within time windows. Consequently, the MRCG model has the potential to facilitate the joint optimization of communication and computation strategies within LEOSNs. In detail, the MRCG model is instrumental in both facilitating problem formulation and its subsequent resolution.

III. Problem Formulation

For clarity, we adopt three steps to introduce the problem formulation as follows.

First, we introduce decision variable $x_{jtc_j}\in\{0,1\}$ to represent the multi-resource allocation strategy, such that $x_{jtc_j}=1$ indicates that task j is compressed with the compression ratio of c_j and then scheduled in time slot t, otherwise $x_{jtc_j}=0$. As such, we can use variable x_{jtc_j} to represent whether any vertex $v\in\mathcal{V}_j$ is selected, so that $x_{jtc_j}=1$ indicates that task j is offloaded with c_j in time slot $t=\eta^{\mathrm{st}}_{jk}(v)$, otherwise $x_{jtc_j}=0$. This suggests a one-to-one correspondence between decision variable x_{jtc_j} and vertex $v\in\mathcal{V}_j$. For clarity, we use $v(x_{jtc_j})$ to represent the vertex v derived from x_{jtc_j} and $x_{jtc_j}(v)$ to denote the variable x_{jtc_j} obtained from v.

Then, we respectively define the maximization of the sum task weights and the minimization of the total data loss as $\max_{\boldsymbol{x}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}_j}\sum_{c_j\in\mathcal{C}_{s(j)}}x_{jtc_j}w_j$ and $\min_{\boldsymbol{x}}\sum_{j\in\mathcal{J}}(1-\sum_{t\in\mathcal{T}_j}\sum_{c_j\in\mathcal{C}_{s(j)}}x_{jtc_j}c_j)D_j$, where $\mathcal{T}_j=\{t|t=\eta^{\mathrm{st}}_{jk}(v),v\in\mathcal{V}_j\}$ denotes the set of the feasible offloading start times.

To balance these two optimization objectives, we adopt the weighted Tchebycheff method [15]–[17] to obtain a normalized weighted combination of them as follows:

$$\max_{\boldsymbol{x}}\ \frac{\theta\sum\limits_{j\in\mathcal{J}}\sum\limits_{t\in\mathcal{T}_j}\sum\limits_{c_j\in\mathcal{C}_{s(j)}}x_{jtc_j}w_j}{w_{\max}|\mathcal{J}|}-\frac{(1-\theta)\sum\limits_{j\in\mathcal{J}}(1-\sum\limits_{t\in\mathcal{T}_j}\sum\limits_{c_j\in\mathcal{C}_{s(j)}}x_{jtc_j}c_j)D_j}{D_{\max}|\mathcal{J}|},\tag{2}$$

where $\boldsymbol{x}=\{x_{jtc_j}\}$, $w_{\max}=\arg\max\limits_{j\in\mathcal{J}}w_j$, $D_{\max}=\arg\max\limits_{j\in\mathcal{J}}D_j$, and $\theta\in[0,1]$ is a weighting factor to balance the the maximization of the sum task weights and the minimization of the total data loss.

We let $\phi = \frac{(1-\theta)}{D_{\max}|\mathcal{J}|}\sum_{j\in\mathcal{J}} D_j$ and rearrange (2) to yield:

$$\max_{\boldsymbol{x}} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}_j}\sum_{c_j\in\mathcal{C}_{s(j)}} \psi_j(\theta)x_{jtc_j} - \phi, \tag{3}$$

where $\psi_j(\theta) = \frac{1}{|\mathcal{J}|}(\frac{\theta w_j}{w_{\max}} + \frac{(1-\theta)c_j D_j}{D_{\max}})$.

Finally, we utilize the MRCG model to formulate the studied problem into the form of LIP below:

$$\begin{aligned}
\textbf{P0}:\ & \max_{\boldsymbol{x}} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}_j}\sum_{c_j\in\mathcal{C}_{s(j)}} \psi_j(\theta)x_{jtc_j} - \phi\\
\text{s.t. } & \text{C1}: \sum_{t\in\mathcal{T}_j}\sum_{c_j\in\mathcal{C}_{s(j)}} x_{jtc_j} \le 1, \forall j\in\mathcal{J},\\
& \text{C2}: x_{itc_i} + x_{jt'c'_j} \le 1, \forall (u(x_{itc_i}), v(x_{jt'c'_j}))\in\mathcal{E},\\
& \quad \forall i\neq j\in\mathcal{J}, t\in\mathcal{T}_i, t'\in\mathcal{T}_j, c_i\in\mathcal{C}_{s(i)}, c_j\in\mathcal{C}_{s(j)},\\
& \text{C3}: x_{jtc_j}\in\{0,1\}, \forall j\in\mathcal{J}, t\in\mathcal{T}_j, c_j\in\mathcal{C}_{s(j)},
\end{aligned}$$

where C1 indicates that any task is offloaded at most once, C2 reflects that the resource allocation conflicts in the model of MRCT, and C3 is the binary constraint.

Remark 2. In real-world implementations, the weighting factor θ can be methodically modulated within the range $[0, 1]$ to obtain a set of pareto optimal solutions to **P0** with respect to total data loss and the sum weights of tasks. The pair of optimization objectives that best matches the desired criteria can then be selected.

Remark 3. In **P0**, the allocation of computation resources for a given task equates to assigning a corresponding compression ratio to that task. Moreover, it is assumed that this allocation process is devoid of any energy consumption.

Remark 4. It is worth observing that a classical time window scheduling problem is nested in **P0**, which have been turned out to be NP-hard. In view of this, we aim to design a low-complexity approximation algorithm for solving **P0**. Furthermore, based on the MRCT model, **P0** is formulated as a special case of LIP, which could be solved by the SDR method efficiently.

IV. Proposed Solution

In this section, we adopt a powerful SDR method [18] to solve **P0** due to its special linear structure. We explore this special linear structure of **P0** and transform it into a non-convex Quadratically Constrained Quadratic Program (QCQP) form. Then, the derived QCQP is transformed into a convex standard form of SDR. Finally, we propose a fast yet efficient algorithm to generate some high-quality solutions for **P0**.

To achieve a QCQP form of **P0**, we first equivalently rewrite C3 as the following constraint:

$$\text{C4}: x_{jtc_j}(x_{jtc_j}-1) = 0, \forall j\in\mathcal{J}, t\in\mathcal{T}_j, c_j\in\mathcal{C}_{s(j)}.$$

Then, we define $\boldsymbol{x} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{\mathcal{J}}]^T$ with each element $\boldsymbol{x}_j = [x_{j11}, ..., x_{j|\mathcal{T}_j||\mathcal{C}_{s(j)}|}]^T$ being the $1\times|\mathcal{T}_j||\mathcal{C}_{s(j)}|$ column vector for all j. Let $\boldsymbol{b}_j = \frac{1}{2}\psi_j(\theta)\mathbf{1}_{O_j\times 1}$ with $O_j = |\mathcal{T}_j||\mathcal{C}_{s(j)}|$. We let $\boldsymbol{e}^n_{O_i\times 1}$ and $\boldsymbol{e}^{n'}_{O_j\times 1}$ respectively denote O_i-dimensional and O_j-dimensional column unit vectors with the nth and n'th elements being one. As such, the one to one mapping exists between $n\in\{1, ..., O_j\}$ and $v\in\mathcal{V}_j$ for any task j. Let $n(v)$ index n mapped from the vertex v.

On the basis of the defined vectors above, we write the vector form of **P0** as follows:

$$\begin{aligned}
\textbf{P1}:\ & \max_{\boldsymbol{x}} \sum_{j\in\mathcal{J}} (\boldsymbol{b}_j)^T\boldsymbol{x}_j - \phi\\
\text{s.t. } & (\mathbf{1}_{O_j\times 1})^T\boldsymbol{x}_j \le 1, \forall j\in\mathcal{J},\\
& (\boldsymbol{e}^{n(u)}_{O_i\times 1})^T\boldsymbol{x}_i + (\boldsymbol{e}^{n'(v)}_{O_j\times 1})^T\boldsymbol{x}_j \le 1, \forall (u,v)\in\mathcal{E}, i\neq j\in\mathcal{J},\\
& (\boldsymbol{x}_j)^T\text{diag}(\boldsymbol{e}^n_{O_j\times 1})\boldsymbol{x}_j - (\boldsymbol{e}^n_{O_j\times 1})^T\boldsymbol{x}_j = 0,\\
& \quad \forall n\in\{1, ..., O_j\}, j\in\mathcal{J},\\
& \boldsymbol{x}_j \ge 0, \forall j\in\mathcal{J}.
\end{aligned}$$

Furthermore, we define $\boldsymbol{q}_j = [(\boldsymbol{x}_j)^T, 1]^T$ and get rid of the constant term of ϕ from **P1** to obtain the QCQP form:

$$\begin{aligned}
\textbf{P2}:\ & \max_{\{\boldsymbol{q}_j\}} \sum_{j\in\mathcal{J}} (\boldsymbol{q}_j)^T\boldsymbol{G}^{\mathcal{G}}_j\boldsymbol{q}_j\\
\text{s.t. } & (\boldsymbol{q}_j)^T\boldsymbol{G}^{\mathcal{A}}_j\boldsymbol{q}_j \le 1, \forall j\in\mathcal{J},\\
& (\boldsymbol{q}_i)^T\boldsymbol{G}^{\mathcal{B}}_i\boldsymbol{q}_i + \boldsymbol{q}_j^T\boldsymbol{G}^{\mathcal{B}}_j\boldsymbol{q}_j \le 1, \forall (u,v)\in\mathcal{E},\\
& (\boldsymbol{x}_j)^T\boldsymbol{G}^{\mathcal{C}}_j\boldsymbol{x}_j = 0, \forall j\in\mathcal{J},\\
& \boldsymbol{x}_j \ge 0, \forall j\in\mathcal{J},
\end{aligned}$$

where

$$\begin{aligned}
&\boldsymbol{G}^{\mathcal{A}}_j = \begin{bmatrix} \mathbf{0} & \frac{1}{2}\mathbf{1}_{O_j\times 1}\\ \frac{1}{2}(\mathbf{1}_{O_j\times 1})^T & \mathbf{0}\end{bmatrix},\\
&\boldsymbol{G}^{\mathcal{G}}_j = \begin{bmatrix} \mathbf{0} & \frac{1}{2}\boldsymbol{b}_j\\ \frac{1}{2}(\boldsymbol{b}_j)^T & \mathbf{0}\end{bmatrix}, \boldsymbol{G}^{\mathcal{B}}_i = \begin{bmatrix} \mathbf{0} & \frac{1}{2}\boldsymbol{e}^{n(u)}_{1\times O_i}\\ \frac{1}{2}(\boldsymbol{e}^{n(u)}_{1\times O_i})^T & \mathbf{0}\end{bmatrix},\\
&\boldsymbol{G}^{\mathcal{B}}_j = \begin{bmatrix} \mathbf{0} & \frac{1}{2}\boldsymbol{e}^{n'(v)}_{1\times O_j}\\ \frac{1}{2}(\boldsymbol{e}^{n'(v)}_{1\times O_j})^T & \mathbf{0}\end{bmatrix},\\
\text{and }\ &\boldsymbol{G}^{\mathcal{C}}_j = \begin{bmatrix} \text{diag}(\boldsymbol{e}^n_{1\times O_j}) & -\frac{1}{2}\boldsymbol{e}^n_{1\times O_j}\\ \frac{1}{2}(\boldsymbol{e}^n_{1\times O_j})^T & \mathbf{0}\end{bmatrix}.
\end{aligned}$$

Solving **P2** directly still has exponential time complexity. Toward this end, we define $\boldsymbol{X}_j = \boldsymbol{q}_j(\boldsymbol{q}_j)^T$ and drop the rank constraint of rank$(\boldsymbol{X}_j), \forall j$, to yield a convex standard SDR form of **P2** as follows:

$$\begin{aligned}
\textbf{P3}:\ & \max_{\{\boldsymbol{X}_j\}} \sum_{j\in\mathcal{J}} \text{Tr}(\boldsymbol{G}^{\mathcal{G}}_j\boldsymbol{X}_j)\\
\text{s.t. } & \text{Tr}(\boldsymbol{G}^{\mathcal{A}}_j\boldsymbol{X}_j) \le 1, \forall j\in\mathcal{J},\\
& \text{Tr}(\boldsymbol{G}^{\mathcal{B}}_i\boldsymbol{X}_i) + \text{Tr}(\boldsymbol{G}^{\mathcal{B}}_j\boldsymbol{X}_j) \le 1, \forall (u,v)\in\mathcal{E},\\
& \text{Tr}(\boldsymbol{G}^{\mathcal{C}}_j\boldsymbol{X}_j) = 0, \forall j\in\mathcal{J},\\
& \boldsymbol{X}_j(O_j+1, O_j+1) = 1, \forall j\in\mathcal{J},\\
& \boldsymbol{X}_j \succeq 0, \forall j\in\mathcal{J}.
\end{aligned}$$

As far, **P3** can be solved optimally in polynomial time complexity via some Semi-Definite Programming (SDP) solvers, such as SeDuMi, SDPNAL, and SDPT3 [19].

We implement the following three steps to efficiently solve **P0**: First, we use one SDP solver to solve **P3** for obtaining an optimal solution $\boldsymbol{X}^* = \{\boldsymbol{X}^*_j\}$. Second, we adopt the Gaussian randomization method to construct a set of approximate solutions to **P3**. In detail, we execute L rounds of randomization

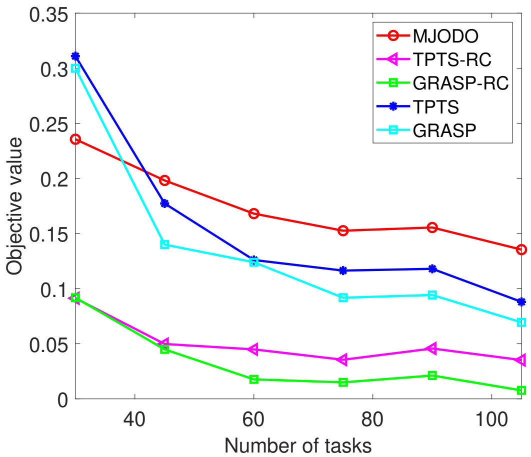

Fig. 3. Objective value versus the number of tasks.

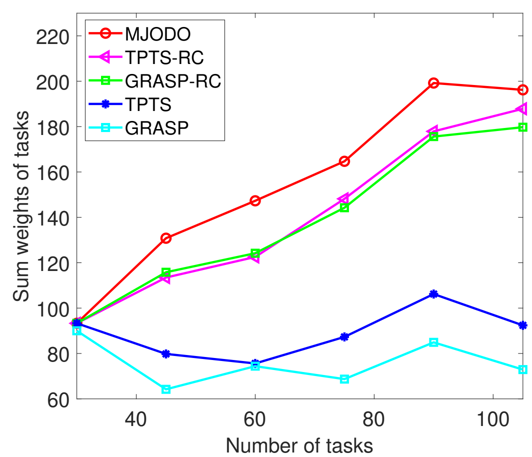

Fig. 4. Sum weights of tasks versus the number of tasks.

Algorithm 1 Proposed MJODO Algorithm

Input: $\mathcal{T}_j$, $\mathcal{C}_{s(j)}$, D_j, R_{jk}, θ, $w_{\max}$, $D_{\max}$.

1: **Initialization:** $\tilde{\boldsymbol{\xi}}^l = \mathbf{0}, \forall l$.
2: Obtain the optimal solution $\boldsymbol{X}^* = \{\boldsymbol{X}_j^*\}$ to **P3** via the SDP solver.
3: **for** $l = 1 : L$ **do**
4:   **for** $j = 1 : J$ **do**
5:     Generate $\boldsymbol{\xi}_j^l \sim \mathcal{N}(0, \boldsymbol{X}_j^*)$.
6:   **end for**
7: **end for**
8: Sort all elements of $\boldsymbol{\xi}^l$ in decreasing order.
9: **for** $j = 1$ to the length of $\boldsymbol{\xi}^l$ **do**
10:   **if** Both C1 and C2 are satisfied **then**
11:     $\tilde{\boldsymbol{\xi}}_j^l = 1$.
12:   **end if**
13: **end for**
14: Obtain $l^* = \arg \max_{l} \sum_{j \in \mathcal{J}} (\boldsymbol{b}_j)^T \tilde{\boldsymbol{\xi}}_j^l$.

Output: $\tilde{\boldsymbol{\xi}}^{l^*}$.

with $\boldsymbol{\xi}_j^l$ denoting the lth approximate solution for task j. For any $l \in \{1, ..., L\}$, we generate the value of $\boldsymbol{\xi}_j^l$ using a normal distribution with a mean of zero and a variance of $\boldsymbol{X}_j^*$, expressed as $\boldsymbol{\xi}_j^l \sim \mathcal{N}(0, \boldsymbol{X}_j^*)$. Let $\boldsymbol{\xi}^l = \{\boldsymbol{\xi}_j^l\}$. Third, we convert $\boldsymbol{\xi}_j^l, \forall l$ into feasible solutions to **P0**, subsequently deriving the best optimization objective. We summarize the above steps into Algorithm 1, termed Multi-resource Joint Optimization Data Offloading (MJODO).

The complexity of the proposed MJODO algorithm is dominated by step 2. In step 2, **P3** can be optimally solved by the SeDuMi solver, whose the complexity is $O(n^{4.5}\log(\frac{1}{\epsilon}))$ with $\epsilon > 0$ denoting the solution accuracy [18]. As such, the total complexity of the proposed MJODO algorithm is $O(n^{4.5}\log(\frac{1}{\epsilon}))$.

V. Numerical Results

The simulation scenario for a LEOSN comprises three LEO satellites and three data receiver antennas. The whole scheduling horizon is set at two hours, with each time slot lasting three minutes. We utilize the dataset from [6] to generate the intermittent time windows between the three LEO satellites and three data receiver antennas. This dataset randomly generates the transmission time of each task within a time window using a probability distribution derived from a real tracking and data relay satellite system. The weight value of each task is generated using a uniformly distributed function within the interval $[1, 5]$. The compression ratio set for any LEO satellite is set to $\{0.3, 0.5, 0.8\}$. The value of θ is set to 0.5. We employ the free MATLAB toolbox YALMIP to solve **P3** with the solver set as SeDuMi.

The four benchmarks for performance comparison with the proposed MJODO are Greedy Randomized Adaptive Search Procedure (GRASP) algorithm [6], GRASP with Random Compression (GRASP-RC), Two-Phase Task Scheduling (TPTS) algorithm [7], and TPTS with Random Compression (TPTS-RC). The GRASP-RC and TPTS-RC schemes assign a single compression ratio from the set of $\{0.3, 0.5, 0.8\}$ to each task at random with equal probability.

In Figs. 3, 4, and 5, we evaluate the proposed MJODO algorithm against four benchmarks based on network metrics: objective value, sum weights of tasks, and total data loss versus the number of tasks. As shown in Fig. 3, it is observed that MJODO substantially outperforms both TPTS-RC and GRASP-RC in terms of objective value as the number of tasks increases. This is because MJODO assigns appropriate compression ratios for tasks according to time windows and the number of tasks. Furthermore, Fig. 3 shows that MJODO achieves a higher objective value than both TPTS and GRASP as the task number grows, with the exception when the task number is exactly 30. This suggests two key insights: Firstly, when the task number is 30 or fewer, there is no need to compress tasks due to the sufficiency of the communication resources of LEOSN; Secondly, integrating computation into LEOSN becomes essential when these communication resources are insufficient to facilitate task scheduling.

From Fig. 4, it is evident that the schemes of TPTS-RC and GRASP-RC yield a higher sum weight of tasks compared to the schemes of TPTS and GRASP. This indicates that

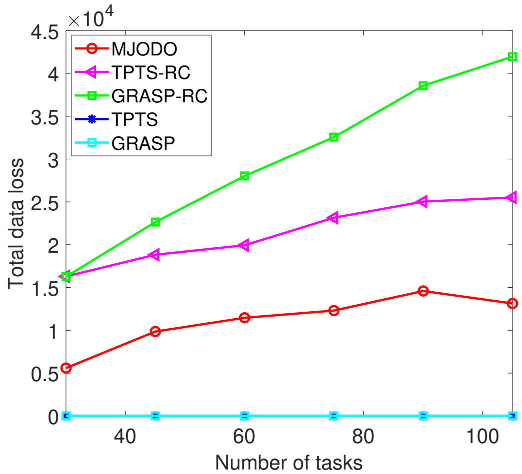

Fig. 5. Total data loss versus the number of tasks.

integrating computation resources into LEOSN for task data compression enhances the sum weights of tasks. Conversely, Fig. 5 shows that while TPTS-RC and GRASP-RC incur significant total data loss, TPTS and GRASP do not experience any data loss. This underscores that random data compression in the both TPTS-RC and GRASP-RC can lead to unnecessary data loss. A closer examination of Figs. 4 and 5 reveals that MJODO achieves higher sum weight of tasks with less total data loss compared to TPTS-RC and GRASP-RC. The underlying rationale is that MJODO concurrently optimizes data compression and task scheduling, aiming to maximize the sum weights of tasks while minimizing unnecessary data loss. This underscores the importance of joint optimization of data compression and task scheduling in LEOSNs.

VI. Conclusion

We have studied the joint data compression and task scheduling problem for LEOSNs to maximize the normalized weighted combination of the sum task weight and the total data loss, while adhering to time window constraints. To facilitate problem formulation and solving, we first propose the novel MRCG model to depict the task transmission conflicts inherent in task transmission during a typical data offloading procedure in LEOSNs. Then, we formulate the studied problem into a standard form of ILP. We further employ the SDR method to propose an efficient solution for solving the formulated ILP. The comprehensive simulation results conclusively demonstrate that the joint data compression and task scheduling can significantly enhance the performance of LEOSNs in terms of both maximizing the task sum weight and minimizing the total data loss. Furthermore, our proposed solution exhibits superior performance compared to existing benchmarks.

References

[1] D. Han, Q. Ye, H. Peng, W. Wu, H. Wu, W. Liao, and X. Shen, "Two-timescale learning-based task offloading for remote IoT in integrated satellite–terrestrial networks," *IEEE IoT J.*, vol. 10, no. 12, pp. 10 131–10 145, Jun. 2023.

[2] H. Guo, Y. Wang, J. Liu, and C. Liu, "Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 347–359, Jan. 2024.

[3] J. Zhang, Q. Ye, K. Qu, Y. Sun, Y. Tang, D. Zhao, and T. Ye, "Learning-assisted dynamic VNF selection and chaining for 6G satellite-ground integrated networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 1, pp. 1504–1519, Jan. 2025.

[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. Feb. 2023.

[5] H. Guo, X. Zhou, J. Wang, J. Liu, and A. Benslimane, "Intelligent task offloading and resource allocation in digital twin based aerial computing networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3095–3110, Oct. 2023.

[6] S. Rojanasoonthon, *Parallel Machine Scheduling with Time Windows. Ph.D. dissertation*, Univ. Texas at Austin, 2004.

[7] 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.

[8] X. Chen, X. Li, X. Wang, Q. Luo, and G. Wu, "Task scheduling method for data relay satellite network considering breakpoint transmission," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 844–857, Jan. 2021.

[9] W. Liu, L. Zhu, H. Yang, H. Li, J. Li, and A. M.-C. So, "Maximum flow routing strategy with dynamic link allocation for space information networks under transceiver constraints," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9993–10 000, Sep. 2022.

[10] F. Wang, D. Jiang, Z. Wang, J. Chen, and T. Q. S. Quek, "Dynamic networking for continuable transmission optimization in LEO satellite networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6639–6653, May 2023.

[11] 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. 18 126–18 141, Dec. 2024.

[12] L. He, K. Guo, H. Gan, and L. Wang, "Collaborative data offloading for earth observation satellite networks," *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1116–1120, May 2022.

[13] P. Wang, H. Li, B. Chen, and S. Zhang, "Enhancing earth observation throughput using inter-satellite communication," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7990–8006, Oct. 2022.

[14] Z. Wang, B. Lin, Q. Ye, Y. Fang, and X. Han, "Joint computation offloading and resource allocation for maritime MEC with energy harvesting," *IEEE IoT J.*, vol. 11, no. 11, pp. 19 898–19 913, Jun. 2024.

[15] D. W. K. Ng, E. S. Lo, and R. Schober, "Multiobjective resource allocation for secure communication in cognitive radio networks with wireless information and power transfer," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3166–3184, May 2016.

[16] H. Guo, X. Chen, X. Zhou, and J. Liu, "Trusted and efficient task offloading in vehicular edge computing networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 10, no. 6, pp. 2370–2382, Dec. 2024.

[17] 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.

[18] Z. Luo, W. Ma, A. M. 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.

[19] Y. Yamakawa, T. Ikegami, E. H. Fukuda, and N. Yamashita, "An equivalent nonlinear optimization model with triangular low-rank factorization for semidefinite programs," *Optim. Method Softw.*, vol. 38, no. 6, pp. 1296–1310, Jul. 2023.