Demand-Aware Onboard Payload Processor Management for High-Throughput NGSO Satellite Systems

TEDROS SALIH ABDU, Member, IEEE
STEVEN KISSELEFF, Senior Member, IEEE
EVA LAGUNAS, Senior Member, IEEE
University of Luxembourg, Luxembourg, Luxembourg

JOËL GROTZ, Senior Member, IEEE
Société Européenne des Satellites (SES), Betzdorf, Luxembourg

SYMEON CHATZINOTAS, Fellow, IEEE
BJÖRN OTTERSTEN, Fellow, IEEE
University of Luxembourg, Luxembourg, Luxembourg

High-throughput satellite (HTS) systems with digital payload technology have been identified as a key enabler to support 5G/6G high-data connectivity with wider coverage area. The satellite community has extensively explored resource allocation methods to achieve this target. Typically, these methods do not consider the intrinsic architecture of the flexible satellite digital payload, which consists of multiple processors responsible for receiving, processing, and transmitting the signals. This article presents a demand-aware onboard processor management scheme for broadband nongeostationary satellites. In this context, we formulate an optimization problem to minimize the number of active onboard processors while meeting the system constraints and user requirements. As the problem is nonconvex, we solve it in two steps. First, we transform the problem into demand-driven bandwidth allocation while fixing the number of processors. Second, using the bandwidth allocation solution, we determine the required number of processors with two methods: 1) sequential optimization with the branch-and-bound method and 2) bin packing with next-fit, first-fit, and best-fit methods. Finally, we demonstrate the proposed methods with extensive numerical results. It is shown that the branch-and-bound, best-fit, and first-fit methods manage the processors better than the next-fit method. Furthermore, branch-and-bound method requires fewer processors than the above methods.

Manuscript received 2 August 2022; revised 12 December 2022; accepted 7 February 2023. Date of publication 15 February 2023; date of current version 11 October 2023.

DOI. No. 10.1109/TAES.2023.3245044

Refereeing of this contribution was handled by M. De Sanctis.

This work was supported by the Luxembourg National Research Fund (FNR) under the Project FlexSAT (C19/IS/13696663), the Project DISBuS, and the AFR Grant INSAT (FNR14603732)—"Power and Bandwidth Allocation for INterference-Limited SATellite Communication Systems".

Authors' addresses: Tedros Salih Abdu, Steven Kisseleff, Eva Lagunas, Symeon Chatzinotas, and Björn Ottersten are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855 Luxembourg, Luxembourg, E-mail: (tedros-salih.abdu@uni.lu; steven.kisseleff@uni.lu; eva.lagunas@uni.lu; symeon.chatzinotas@uni.lu; bjorn.ottersten@uni.lu); Joël Grotz is with Société Européenne des Satellites (SES), Betzdorf 6815, Luxembourg, E-mail: (Joel.Grotz@ses.com). *(All authors contributed equally to this work.) (Corresponding author: Tedros Salih Abdu.)*

I. INTRODUCTION

Satellite communication network providers are expected to offer broadband connectivity to meet the needs of an increasingly heterogeneous market, including the broadcast industry, airplane industry, maritime sector, government agencies, and end-users [1]. Furthermore, satellite technology has a significant role in the era of 5G and beyond in terms of integrating satellite networks with terrestrial networks, providing backhaul services, and offering coverage for the Internet-of-Things (IoT) applications and beyond [2], [3], [4].

Satellites are now incorporating advanced digital payload technologies to adapt to these diverse markets and emerging applications. These payloads have the reconfigurable capability to perform various functions, including changing the beam coverage, allocating satellite resources, and adjusting the Radio-Frequency (RF) power distribution dynamically in response to traffic demands [5], [6]. However, advanced management techniques must be employed to better exploit the functionality of these digital payloads.

Several studies have been conducted regarding satellite resource management based on user demand requirements. In this case, the power allocation over satellite downlinks in light of traffic demand and channel characteristics has been investigated in [7]. Moreover, in [8], power allocation has been proposed while considering interbeam interference among the beams. Additionally, a two-stage power allocation method using metaheuristics to minimize the system's unmet capacity has been proposed in [9]. Similarly, an energy-aware power-allocation technique to minimize unmet system capacity and transmit power consumption has been explored in [10]. Furthermore, a reinforcement-based approach and a game-based approach for allocating the system power have been developed in [11], [12] and [13], [14], respectively. However, these approaches only focus on power management without considering payload management and bandwidth allocation. Alternatively, bandwidth allocation is explored in [15] and [16] to match the traffic demand with the system capacity. However, power allocation and payload management are not incorporated into these methods.

In the literature, different resource allocation methods have been discussed with regard to joint power and bandwidth allocation. In [17], a modified version of the simulated annealing algorithm has been proposed in order

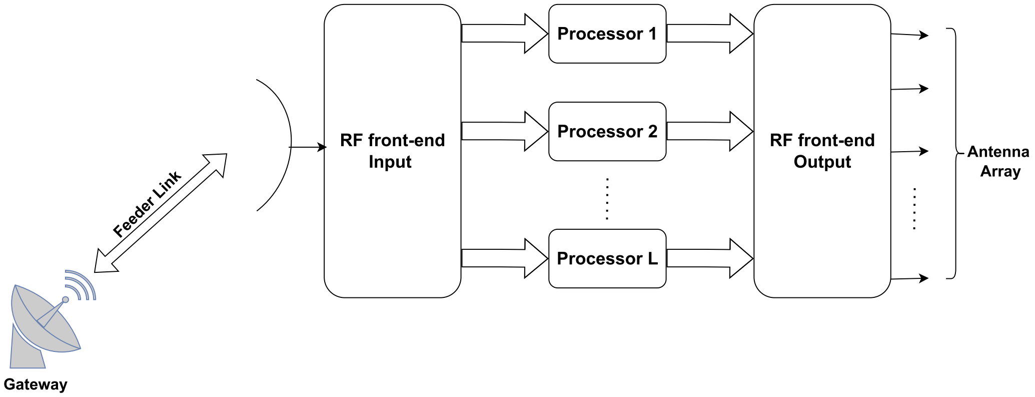


Fig. 1. Generic digital payload system.

to achieve a fair distribution of power and bandwidth across beams while simultaneously taking fairness into account. Similarly, reinforcement-based techniques in [18], [19], and [20] and using conventional neural networks in [21] have been proposed to jointly allocate the power and the bandwidth of the satellite system. Furthermore, an iterative convex optimization approach to utilize the satellite resource efficiently has been studied in [22], [23], and [24]. Additionally, resource optimization for integrating satellite-terrestrial communication has been considered in [25], [26], [27], [28], [29], and [30].

There have been a few studies that have examined the relationship between the management of resources and the intrinsic architecture of digital payloads. A model for quantitatively evaluating the flexibility of payloads with a digital channelizer has been explored in [31]. Additionally, in [32], a frequency resource allocation method based on beam requests has been proposed to improve the overall throughput of satellite communication systems. Moreover, it investigates the advantages of digital channelizers compared with the traditional resource allocation method.

While the above methods manage the satellite resources, they do not address the constraints imposed by the payload processors, which require a different perspective. Fig. 1 illustrates a generic digital payload comprising RF inputs, RF outputs, and digital processor ports equipped with Analog-to-Digital Converters (ADCs), Digital-to-Analog Converters (DACs), filters, and modulation/demodulation techniques. A typical payload of this type is called a regenerative onboard processor. Alternatively, it is known as a digital transparent processor if the payload does not provide modulation and demodulation capabilities [33].

The RF front-end input is used to down-convert the received signal into a baseband/intermediate frequency (IF). It consists of a filter, a low-noise amplifier, a mixer, and a gain controller. The processor then converts the baseband/IF signal to digital format using ADC and applies channelization (filters), modulation, and demodulation techniques to process the signal further. Finally, the digital signal is converted back to an analog signal through DAC. The DAC output signal is filtered and amplified on the RF front-end output side for transmission [5], [33].

The payload[1] operates the signals through multiple digital processors because 1) a single processor can handle only a limited number of beams in order to reduce signal processing complexity and to avoid unnecessary delays due to sequential processing of tasks, and 2) two or more signals with the same carrier frequency need independent processors to avoid interference. Therefore, appropriate mapping of the carrier bandwidth of signals to processors is required. On the other hand, the optimized signal bandwidth may depend on the user demand; a low demand requires less bandwidth, whereas a high demand requires a larger bandwidth. Thus, the number of operating processors may vary according to the demand. In general, if the demand is low, fewer processors are needed while more processors become necessary as demand increases. Hence, depending on the demand, not all of the available processors may need to be switched ON. This reduces the amount of power needed to operate the processors in order to configure each beam.

In this context, we focus on the optimization of the forward link bandwidth for broadband nongeostationary (NGSO) satellites with the main goal to minimize the number of active onboard processors while meeting the system constraints and end-user traffic demands. The detailed contributions of this article are summarized as follows.

1) We propose a mathematical formulation for the demand-aware onboard payload processor optimization to minimize the number of operating processors while considering the following aspects: a) frequency planning constraints; b) user demand constraint, and c) processor abstraction constraints. Hence, the satellite payload processors can be managed flexibly according to the system constraint and the beam demand requirement.
2) The optimization problem is nonconvex due to nonlinear functions and integer-valued variables present in the formulation. We propose a two-step approach to efficiently tackle the problem. First, we design

[1] Note that flexible digital payloads have been introduced recently. This is why the optimization for mapping data processing tasks for the individual beams onto the processors has not been considered so far.

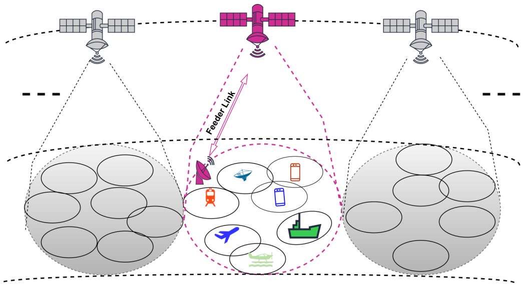


Fig. 2. NGSO constellations with multibeam technology.

the bandwidth allocation strategy based on the frequency planning and demand satisfaction requirements while fixing the number of processors. Second, we consider the processor abstraction problem to determine the minimum number of processors required to accommodate the above bandwidth allocation strategy. We propose the following solutions to address this problem: a) sequential optimization based on the branch-and-bound technique; b) bin packing [34] based on next-fit, first-fit, and best-fit methods.

3) Finally, we demonstrate the performance of the proposed methods through extensive numerical evaluations. We observe that the branch-and-bound, best-fit, and first-fit methods have better performance in flexibly managing the payload processors than the next-fit method.

The rest of the article is organized as follows. In Section II, the system model for the NGSO satellites is provided. The proposed demand-aware onboard payload processor management is presented in Section III. In Section IV, the simulation results are discussed. Finally, Section V concludes the article.

II. SYSTEM MODEL

We consider a downlink NGSO satellite to provide high data-rate connectivity to ground users. To continuously serve the globe, Fig. 2 shows a single constellation with multiple satellites. Since each NGSO satellite possesses similar characteristics, this model focuses on a snapshot of one NGSO satellite within a specific geographic region. This satellite employs multibeam technology to produce N narrow beams to serve a particular geographical area, see Fig. 2. The beams can be generated using direct radiating arrays or focal-array-fed-reflectors-based antennas [35]. The generated beams are assumed to be sparse to reflect a realistic scenario in which the satellite covers only the areas of interest. The O3b mPOWER satellite, for example, uses steerable and shapable spot beams that are continuously shifted and scaled to cover specific geographical areas [36]. Hence, depending on the coverage area, some beams may not have overlapping regions. Furthermore, the desired area covered by the satellite may represent maritime, aeronautical, and terrestrial providers. Therefore, the demand may substantially vary from one beam to another.

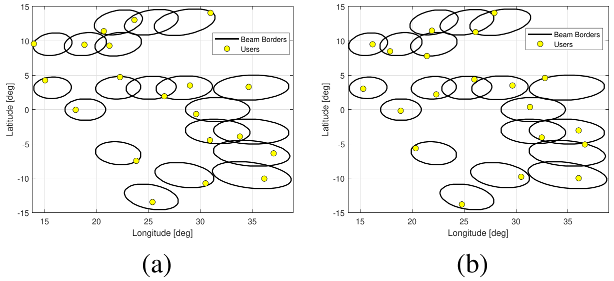


Fig. 3. $N = 20$ beam scenario with an example of user location selection.

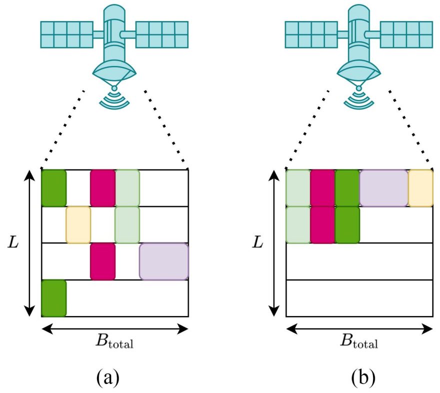


Fig. 4. Processor management. (a) Wasteful configuration. (b) Optimized configuration.

Although multiple users may be served within the same beam coverage area, in this work, we follow a user scheduling abstraction, i.e., we consider a single user per beam and we examine the system's performance at different user locations. For this, we select the user's location from a uniform distribution within the beam's coverage area. For example, Fig. 3(a) and (b) shows the user selection at different instances. Then, the system response for each instance will be recorded and averaged to determine the overall system performance. This will be presented in the simulation results in Section IV. Henceforth, we will use the terms beam and user interchangeably.

A. Satellite Payload Processor Model

The processor is responsible for receiving and processing the signals before forwarding them to the user terminals. Using multiple processors, the satellite payload can process a total bandwidth of B_{fwd} on the forward link. Each processor operates in a *Ka*-band available bandwidth B_{ava} from 19.7 to 20.2 GHz. In this case, $B_{\text{fwd}} \gg B_{\text{ava}}$. In conventional satellite systems, the bandwidth B_{ava} is divided into bandwidth chunks, and a frequency reuse scheme (e.g., a four-color scheme) is used to reuse each bandwidth in adjacent beams. Accordingly, the processors handle the bandwidth chunks. However, bandwidth fragmentation may occur within each processor's available B_{ava}, as shown in Fig. 4(a). This is because conventional systems do not optimize the bandwidth distribution across processors, resulting in the use of a substantial number of processors. Hence, optimizing bandwidth distribution across the processors

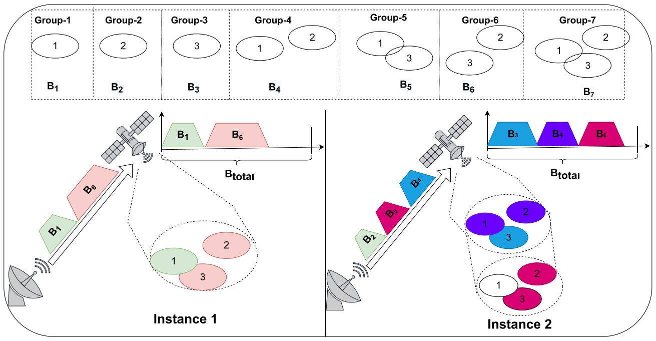


Fig. 5. System bandwidth allocation.

would be more beneficial in minimizing the number of active processors and bandwidth fragmentation. This can be seen in Fig. 4(b) where the bandwidth chunks are rearranged to be handled by fewer processors than in Fig. 4(a). In this context, optimizing bandwidth distribution across the processors will be addressed in Section III.

B. Frequency Planning Model

In our model, each beam can be allocated an orthogonal carrier to avoid cochannel interference. However, there is the possibility of reusing the same carrier frequency band if beams are sufficiently separated. In this case, a group of beams may share the same carrier frequency band. Additionally, beams belonging to the same group require a different processor to process each beam's signal, see Section III-A. In this context, we define the set of groups as $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m, \ldots, \mathcal{G}_M\}$, where $\mathcal{G}_m$ is the mth group of beams and the maximum number of groups is given by $M = 2^N - 1$. Furthermore, we define B_m as the bandwidth of the carrier assigned to the mth group. Fig. 5 shows an example of a satellite system with $N = 3$ beams at different instances. In this case, we have $M = 2^N - 1 = 7$ possible groups and the groups are $\mathcal{G}_1 = \{1\}$, $\mathcal{G}_2 = \{2\}$, $\mathcal{G}_3 = \{3\}$, $\mathcal{G}_4 = \{1, 2\}$, $\mathcal{G}_5 = \{1, 3\}$, $\mathcal{G}_6 = \{2, 3\}$, and $\mathcal{G}_7 = \{1, 2, 3\}$. Accordingly, for each group, the satellite allocates the corresponding bandwidth B_m from the available bandwidth B_{ava}. For example, from Fig. 5, at instance one, the satellite allocates B_1 and B_6 for $\mathcal{G}_1$ and $\mathcal{G}_6$, respectively, while the bandwidth allocation for the other groups is zero. At instance two, the satellite selects $\mathcal{G}_3$, $\mathcal{G}_4$, and $\mathcal{G}_6$ with their corresponding bandwidth B_3, B_4, and B_6. Hence, in our model, each beam can be assigned to multiple carriers. However, the bandwidth allocation must satisfy the following constraints:

$$\begin{aligned} \mathcal{T}1 &: \sum_{\mathcal{G}_m \in \mathcal{G}} B_m \leq B_{\text{ava}} \\ \mathcal{T}2 &: B_m \geq 0 \quad \forall_m. \end{aligned} \tag{1}$$

The constraint $T1$ guarantees that overall bandwidth utilization should not exceed the total available processor bandwidth. The $T2$ constraint implies that B_m cannot be negative. Furthermore, we denote the transmitted power of beam i belonging to the mth group as $p_{i,m}$. The overall power allocation is restricted by the total system power P_{total} and is defined as

$$\begin{aligned} \mathcal{T}3 &: \sum_{i=1}^{N} \sum_{\substack{i \in \mathcal{G}_m, \\ \mathcal{G}_m \in \mathcal{G}}} p_{i,m} \leq P_{\text{total}} \\ \mathcal{T}4 &: p_{i,m} \geq 0 \quad \forall i, m. \end{aligned} \tag{2}$$

Note that the power sharing enabled by flexible payloads has been assumed, such that total power can be arbitrarily distributed across beams [37].

C. Channel Model

The satellite channel is primarily affected by the Line-of-Sight (LoS) component, the antenna radiation pattern, and the rain attenuation [38]. Hence, we can model the channel as

$$h_{i,m}[j] = \sqrt{\frac{G(\theta_{i,m}[j])}{\ell_{\text{FSPL}}[i]\ell_{\text{rain}}[i]}} e^{-\mathrm{j}\phi_i} \tag{3}$$

where $h_{i,m}[j]$ is the channel coefficient from a beam j to beam i of in the mth group and consists of the following.

1) Free-Space Path Loss (FSPL): The attenuation of a signal as it propagates through free space [39]. This path loss is defined as

$$\ell_{\text{FSPL}}[i] = \left(4\pi \frac{d_i}{\lambda}\right)^2 \tag{4}$$

where λ is the carrier wavelength and d_i is the LoS distance between the satellite and the ith beam.

2) Antenna gain (AG): This refers to the overall gain obtained from the receiver and transmitter antennas, which is modeled [40], [41] as follows:

$$\begin{aligned} &G(\theta_{i,m}[j]) \\ &= G_{\text{R}} G_{\max} \left(\frac{J_1\left(u_{i,m}[j]\right)}{2u_{i,m}[j]} + 36 \frac{J_3(u_{i,m}[j])}{(u_{i,m}[j])^3} \right)^2 \end{aligned} \tag{5}$$

where G_{R} is the receiver gain, $G_{\max}$ is the maximum beam gain, and $J_1(u_{i,m}[j])$ and $J_3(u_{i,m}[j])$ are first-order and third-order Bessel functions of the first kind, respectively. Furthermore,

$$u_{i,m}[j] = 2.07123 \frac{\sin\left(\theta_{i,m}[j]\right)}{\sin\left(\theta_{3\text{ dB}}\right)}$$

where $\theta_{i,m}[j]$ represents the angle between the center of the beam j and the desired location of beam i as viewed from the satellite. Furthermore, $\theta_{3\text{-dB}}$ refers to the half-power beamwidth.

3) Rain attenuation: This reduces the amplitude of the transmitted signal by scattering and absorbing it. This rain attenuation effect is notably severe for carrier frequencies above 10 GHz. Following the recommendation of TU-R P.618-13 [42], ITU-R P.839 [43], ITU-R P.837 [44], and ITU-R P.838 [45], the rain attenuation is given by

$$\ell_{\text{rain}}[i] = 10^{\frac{A_i[\mathcal{P}]}{10}} \tag{6}$$

where $A_i[\mathcal{P}]$[dB] is the rain attenuation for beam i with a percentage $\mathcal{P}$ of the average rainfall rate in a year [44].

Let $\mathbf{g}_m[i] = \left[g_{i,m}[1], g_{i,m}[2], \ldots, g_{i,m}[|G_m|]\right]^T$ with $g_{i,m}[j] = |h_{i,m}[j]|^2$ represents the channel gain vector of beam i belonging to the mth group, where $|\mathcal{G}_m|$ is the cardinality of the mth group. While having the transmit power and channel definitions, the Signal-to-Interference-plus-Noise Ratio (SINR) of beam i in the mth group is determined as

$$\gamma_m[i] = \frac{g_{i,m}[i]p_{i,m}}{\sum_{j=1,\,j\neq i}^{|\mathcal{G}_m|} g_{i,m}[j]p_{j,m} + N_0 B_m}. \tag{7}$$

Then, the Shannon capacity for the beam i in the mth group is

$$C_m[i] = B_m \log_2(1 + \gamma_m[i]). \tag{8}$$

Subsequently, the total capacity provided by the system to the beam i is

$$C[i] = \sum_{\substack{i\in\mathcal{G}_m \\ \mathcal{G}_m\in\mathcal{G}}} C_m[i]. \tag{9}$$

This capacity should match the demand $D[i] > 0$, i.e.,

$$\mathcal{T}5 : C[i] \geq D[i] \quad \forall_i. \tag{10}$$

However, for larger demands, the constraint $\mathcal{T}5$ may not feasible to satisfy. In this case, the total demand satisfaction in all beams is measured by the unmet system capacity as follows:

$$C_{\text{unmet}} = \sum_{i=1}^{N} \max\left(1 - \frac{C[i]}{D[i]}, 0\right). \tag{11}$$

III. PROPOSED DEMAND-AWARE ONBOARD PAYLOAD PROCESSOR MANAGEMENT

In this section, we propose a demand-aware onboard payload processor management optimization for high throughput NGSO satellite systems. First, the processor abstraction will be presented to provide details on how the processor will handle the beam bandwidth chunks. Then, we formulate a problem while considering the satellite payload model, the frequency planning model, the channel model, and the processor abstraction. Afterward, a solution will be presented for this problem.

A. Processor Abstraction

This section discusses in detail the system constraints associated with processor abstraction. A single processor must handle orthogonal carriers to avoid signal interference. Hence, signals with the same carrier frequency require different processors. Additionally, a beam may belong to several groups depending on its demand. Correspondingly, multiple bandwidth chunks can be assigned to the same beam.[2] In this case, it is preferable to allocate the bandwidth chunks belonging to the same beam, to the same processor. Consequently, we can avoid the amount of signal processing required to recombine bandwidth chunks from multiple processors for the respective beams. In this context, we provide the specific characteristics of the processor abstraction as follows.

1) *Bandwidth Mapping:* The system is required to assign the corresponding bandwidth chunks of beams to active processors. Let $x_m[i][l]$ be a binary value mapping indicator with $x_m[i][l] = 1$ indicating that the bandwidth chunk B_m of beam i belonging to the mth group is mapped to processor l. Note beams belonging to the mth group share the same bandwidth chunk B_m. Hence, each beam requires different processors to avoid signal interference. On the other hand, beams belonging to different groups can be assigned to the same processor. Furthermore, the system must avoid assigning two or more processors to handle the same bandwidth chunk B_m of beam i. Additionally, mapping is permitted only when the lth processor is in active mode, i.e., $y_l = 1$ otherwise $y_l = 0$ indicating that the lth processor is offline. Hence, we introduce the following constraints:

$$\begin{aligned}
&\mathcal{T}6 : \sum_{\substack{i\in\mathcal{G}_m, \\ \mathcal{G}_m\in\mathcal{G}}} x_m[i][l] \leq 1 \quad \forall_{m,l} \\
&\mathcal{T}7 : \sum_{l=1}^{L} x_m[i][l] \leq 1 \quad \forall_{m,i} \\
&\mathcal{T}8 : |G_m| \min\{1, \lceil B_m/B_{\text{ava}} \rceil\} \leq \sum_{l=1}^{L} \sum_{\substack{i\in\mathcal{G}_m, \\ \mathcal{G}_m\in\mathcal{G}}} x_m[i][l] \quad \forall_m \\
&\mathcal{T}9 : x_m[i][l] \in \{0, 1\} \quad \forall_{m,i,l} \\
&\mathcal{T}10 : y_l \in \{0, 1\} \quad \forall_l \\
&\mathcal{T}11 : x_m[i][l] \leq y_l \quad \forall_{m,l}.
\end{aligned} \tag{12}$$

Here, $\mathcal{T}6$ prevents beams belonging to the same group from accessing the same processor, whereas $\mathcal{T}7$ avoids repetitive bandwidth mapping to the processors for ith beam belonging to mth group. For $B_m > 0$, the constraint $\mathcal{T}8$ ensures that all beams in the mth group are mapped to their respective processors. Note that the symbol $\lceil\cdot\rceil$ in $\mathcal{T}8$ denotes the ceiling function. This mapping is only possible if y_l is equal to one. Furthermore, all the bandwidth chunks mapped to the lth processor must not exceed B_{ava}. Hence, $F_l \leq B_{\text{ava}}$ holds and expressing F_l in terms of $x_m[i][l]$ and B_m as

$$F_l = \sum_{i=1}^{N} \sum_{\substack{i\in\mathcal{G}_m, \\ \mathcal{G}_m\in\mathcal{G}}} x_m[i][l] B_m \tag{13}$$

leads to the following constraint:

$$\mathcal{T}12 : \sum_{i=1}^{N} \sum_{\substack{i\in\mathcal{G}_m, \\ \mathcal{G}_m\in\mathcal{G}}} x_m[i][l] B_m \leq B_{\text{ava}}. \tag{14}$$

2) *Carrier Contiguity:* For practical reasons, we assume that carrier contiguity (CC) can be employed to assign the bandwidth chunks belonging to the same beam to the

[2] In this article, we assume that a single super user per beam represents the aggregate beam demand. Hence, we focus on beam-level bandwidth allocation while user-level subcarrier allocation design is beyond this article's scope.

same processor. This reduces the additional signal processing required by the system to recombine bandwidth chunks belonging to the same beam from multiple processors for transmission. Therefore, we define a binary-valued variable $z_l[i]$ with $z_l[i] = 1$, indicating that all the bandwidth chunks of beam i are mapped to the lth processor. Accordingly, we write this as follows:

$$\begin{aligned}
&\mathcal{T}13 : \sum_{\substack{i\in\mathcal{G}_m,\\ \mathcal{G}_m\in\mathcal{G}}} x_m[i][l] = z_l[i] \sum_{\substack{i\in\mathcal{G}_m,\\ \mathcal{G}_m\in\mathcal{G}}} \min\{\lceil B_m/B_{\text{ava}}\rceil, 1\} \quad \forall_{i,l}\\
&\mathcal{T}14 : \sum_{l=1}^{L} z_i[l] \leq 1 \quad \forall_i\\
&\mathcal{T}15 : z_l[i] \in \{0, 1\} \quad \forall_{i,l}
\end{aligned} \tag{15}$$

where $\mathcal{T}13$ confirms that all the bandwidth chunks are assigned to the lth processor. Furthermore, $\mathcal{T}14$ prevents mapping CC of the ith beam into multiple processors.

B. Problem Formulation

Here, a payload processor management problem is formulated to determine the minimum number of processors needed for onboard signal processing to handle the bandwidth chunks of each beam. Hence, we can closely match the system's capacity with the beam demand and switch OFF completely unused processors to preserve battery power. Additionally, since it is an NGSO satellite, the situation may change many times during the passage. Hence, to improve the adaptability of the obtained solution to future changes, especially since the size of the bandwidth chunk to be added in the future is unknown, it is reasonable to save as much processing capacity as possible. Accordingly, we propose to load the active processors as much as possible, such that some processors would have large capacity savings, i.e., large bandwidth chunks can be accommodated in the future. In this context, we consider to minimize the $\sum_{l=1}^{L} y_l$ while utilizing the maximum load of each active processors, i.e., F_l. Hence, the problem is formulated as follows:

$$\begin{aligned}
&\underset{\substack{y_l, z_i[l], x_m[i][l],\\ B_m, p_{i,m} \forall_{i,l,m}}}{\texttt{minimize}} \quad \sum_{l=1}^{L} w_l \frac{y_l}{F_l + 1}\\
&\texttt{s.t.} \qquad\qquad\quad \mathcal{T}1\text{-}\mathcal{T}15
\end{aligned} \tag{16}$$

where $\sum_{l=1}^{L} w_l = 1$, $w_l < w_{l+1}$, and w_l is the priority factor of the lth processor. Hence, a processor with the lowest priority factor will have a greater chance of being activated than one with the highest priority factor. The objective of the problem (16) and the constraints $\mathcal{T}5$, $\mathcal{T}8$, $\mathcal{T}12$, and $\mathcal{T}13$ have nonlinear functions while the remaining constraints have integer linear functions. Hence, the type of the problem is a mixed-integer nonlinear program. The solution to this problem is thus difficult to obtain for the following reasons.

1) *Nonconvexity*: The nonlinear function of the objective, the constraints $\mathcal{T}5$, $\mathcal{T}8$, $\mathcal{T}12$, and $\mathcal{T}13$ makes the problem nonconvex; thus, convex optimization methods cannot solve it. Hence, (16) needs to be convexified first.

2) *Complexity*: Two factors contribute to the complexity of this optimization: a) The total number of bandwidth and transmit power optimization variables are increases exponentially as the number of beams increases, which is given by $2^N - 1$ and $N(2^N - 1)$, respectively. Thus, as N increases, the computational time for the optimization increases. b) The search space due to the combinatorial type of optimization variables $x_m[i][l]$, y_l, and $z_l[i]$ increase exponentially as N and L increase with $2^{(2^N+N+L-1)}$, 2^L, and 2^{L+N}, respectively. Hence, the computational complexity of (16) combined with its nonconvexity makes the problem much more difficult to solve. For this, complexity reduction is required.

To address the above issues, we decompose the original problem (16) into two parts. First, we solve the frequency planning problem so that the offered capacity closely matches the beam demand while fixing the number of processors. Consequently, we simplify it into a demand-driven bandwidth allocation problem by convexification and complexity reduction. Then, we consider the processor abstraction problem by fixing bandwidth allocation. For this, we propose two methods: 1) Sequential optimization, which uses the branch-and-bound technique and considers the processors sequentially for handling bandwidth allocation. 2) Bin packing, which uses the processors as bins and bandwidth chunks as items to be packed heuristically. In this context, we explore next-fit, first-fit, and best-fit bin-packing methods.

C. Demand-Driven Bandwidth Allocation

Here, we want to allocate the necessary bandwidth and transmit power to each beam based on its demand. Hence, matching the offered capacity with per beam demand is required. For this, we minimize the system unmet capacity in (11) while fixing the number of processors to $\mathcal{L}$, $1 \leq \mathcal{L} \leq L$. For $\mathcal{L}$ processor, we only consider groups of beams containing $\mathcal{L}$ or less beams. In this case, we need only groups from $\mathcal{G}$ that satisfies $\mathcal{G}_n \in \mathcal{Z}$, where $n \in \mathcal{V}$, $\mathcal{V} = \{m : |\mathcal{G}_m| \leq \mathcal{L}, \forall_m\}$ and $\mathcal{Z} = \{\mathcal{G}_n\}, \forall_n$. For example, with $\mathcal{L} = 2$, groups that satisfy $|G_n| \leq 2$ are selected. In this case, if $|G_n| = 2$ is chosen, i.e., two beams share the same B_n, then it is possible to assign one processor to each beam. Accordingly, the problem (16) for $\mathcal{L}$ processors is simplified to the following subproblem:

$$\begin{aligned}
&\underset{B_n, p_{i,n} \forall_{i,n}}{\texttt{minimize}} \quad \sum_{i=1}^{N} \max\left(1 - \frac{C[i]}{D[i]}, 0\right)\\
&\texttt{s.t.} \qquad\quad \mathcal{T}1 : \sum_{\mathcal{G}_n\in\mathcal{Z}} B_n \leq B_{\text{ava}}\\
&\qquad\qquad\quad\ \mathcal{T}2 : B_n \geq 0 \quad \forall_n\\
&\qquad\qquad\quad\ \mathcal{T}3 : \sum_{i\in G_n} \sum_{G_n\in\mathcal{Z}} p_{i,n} \leq P_{\text{total}}\\
&\qquad\qquad\quad\ \mathcal{T}4 : p_{i,n} \geq 0 \quad \forall i, n
\end{aligned} \tag{17}$$

where $C[i] = \sum_{G_n \in \mathcal{Z}, i \in G_n} C_n[i]$. Note that our goal is to closely match the per beam demand with the offered capacity. Hence, the constraint $\mathcal{T}5$ is equivalently represented in the objective function of (17) as unmet system capacity of (11). Unfortunately, the nonlinearity of the SINR makes the (17) nonconvex. Furthermore, it has exponential complexity because the optimization variable. Hence, to tackle the problem, we consider a suboptimal solution of (17). For this, we assume that the spectral power density S_{psd} is known for each beam. This will help us to avoid the nonlinearity of (7). Accordingly, with $p_{i,n} = S_{\text{psd}} B_n$, (7) is rewritten as

$$\gamma_n[i] = \frac{g_{i,n}[i] S_{\text{psd}} B_n}{\sum_{j=1, j\neq i}^{|G_n|} g_{i,n}[j] S_{\text{psd}} B_n + N_0 B_n} = \frac{g_{i,n}[i] S_{\text{psd}}}{\sum_{j=1, j\neq i}^{|G_n|} g_{i,n}[j] S_{\text{psd}} + N_0}. \tag{18}$$

Then, replacing the max function of the objective function using upper bound slack variable ζ_i for $1 - \frac{C[i]}{D[i]} \leq \zeta_i$ and $\zeta_i \geq 0$, the suboptimal of (17) is provided as

$$\begin{aligned} \underset{B_n, \forall_n}{\texttt{minimize}} \quad & \sum_{i=1}^{N} \zeta_i \\ \texttt{s.t.} \quad & \mathcal{T}1, \mathcal{T}2 \\ & \mathcal{T}3 : \sum_{\mathcal{G}_n \in \mathcal{Z}} |\mathcal{G}_n| S_{\text{psd}} B_n \leq P_{\text{total}} \\ & \widetilde{\mathcal{R}}6 : 1 - \frac{C[i]}{D[i]} \leq \zeta_i \quad \forall_i \\ & \widetilde{\mathcal{R}}7 : \zeta_i \geq 0 \quad \forall_i. \end{aligned} \tag{19}$$

Problem (19) is a linear program that can be solved by well-known solvers [46]. Note that this optimization part does not include processor abstraction such as bandwidth mapping and CC. Hence, $\mathcal{L}$ in this optimization reflects the number of operating processors without CC, i.e., a beam's bandwidth chunk can be distributed across multiple processors. However, the exact number of processors required by the system to support CC will be determined in Section III-D. Algorithm 1 shows the demand-driven bandwidth allocation. With the increasing number of processors, the total amount of bandwidth chunks that can be accommodated increases monotonically as well. However, if the number of processors is too low, the bandwidth allocation problem may be infeasible. Hence, the minimum number of processors is determined by solving the bandwidth allocation problem with a single processor and then gradually incrementing their number until the problem becomes feasible. Hence, the algorithm solves and updates the value of $\mathcal{L}$ until the objective function of (19) satisfies the threshold value of ϵ. The ϵ represents the minimum threshold required to match a beam's demand with the capacity offered. In this case, ϵ is selected in the order of 10^{-6} b/s.

Algorithm 1: Demand-driven Bandwidth Allocation.

```
Input: $\mathcal{L} \leftarrow 1$;
while $\sum_{i=1}^{N} \zeta_i \geq \epsilon$ and $\mathcal{L} \leq L$ do
    Solve (19)
    $\mathcal{L} \leftarrow \mathcal{L} + 1$
```

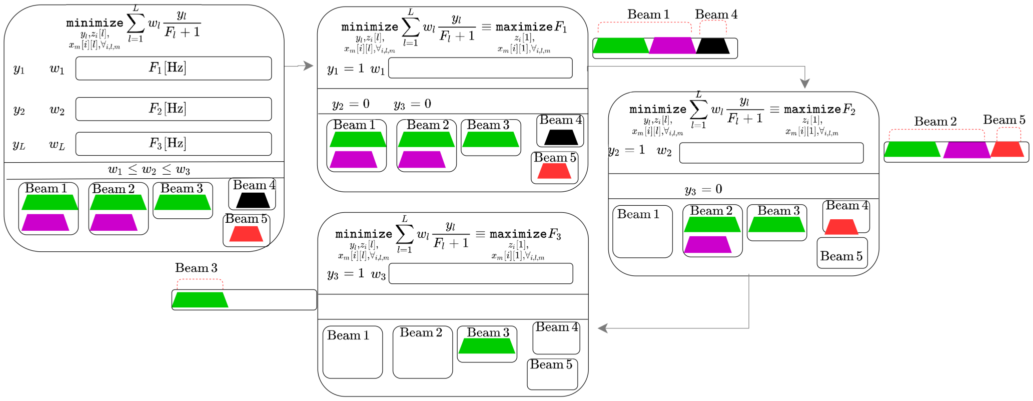
D. Mapping Bandwidth to Processor

In this section, we are interested in mapping the bandwidth of each beam obtained in Section III-C to the processors. With this bandwidth allocation solution, we simplify (16) to the following optimization problem:

$$\begin{aligned} \underset{\substack{y_l, z_i[l], \\ x_n[i][l], \forall_{i,l,n}}}{\texttt{minimize}} \quad & \sum_{l=1}^{L} w_l \frac{y_l}{F_l + 1} \\ \texttt{s.t.} \quad & \mathcal{T}6 - \mathcal{T}15. \end{aligned} \tag{20}$$

Note that $\mathcal{G}_n \in \mathcal{Z}$ and B_n is known, see Section III-C. Hence, $\mathcal{T}6 - \mathcal{T}15$ constraints are updated accordingly. Since the bandwidth B_n is known, the constraints $\mathcal{T}8$, $\mathcal{T}12$, and $\mathcal{T}13$ become linear. However, the problem remains challenging because nonlinearity of the objective function and the aforementioned exponential complexity associated with the combinatorial nature of the integer part of the problem. To tackle this challenge, we propose two methods: 1) sequential optimization and 2) bin packing.

1) *Sequential Optimization Method:* We transform (20) into a sequential optimization problem to solve it. Fig. 6 provides a toy example of sequential optimization that shows a mapping of the bandwidth of 5 beams into three processors. Initially, all processors are in offline mode, thus $y_l = 0, \forall_l$. The first step in mapping bandwidth chunks is to choose which of the three processors to use. Then, we would like to assign as many bandwidth chunks as possible to this specific processor before moving on to the next. Consequently, we can utilize fewer processors. In this context, the processor with the lowest priority factor is selected first. This is because from (20), a processor with a lower priority factor will have a greater chance of being considered for operation than one with a higher priority factor. In this case, w_1 is the lowest priority factor, then the first processor is chosen, i.e., y_1 becomes one while $y_l, \forall_{l \neq 1}$ of the others remains zero. Thus, the minimization of $\sum_{l=1}^{L} w_l \frac{y_l}{F_l + 1}$ is simplified to a minimization of $w_1/(F_1 + 1)$. It is possible to convert minimization of $w_1/(F_1 + 1)$ into maximization by interchanging its numerator and denominator functions, i.e., maximization of $(F_1 + 1)/w_1 \approx F_1$. Accordingly, we maximize F_1 subject to the beam's bandwidth chunks. In this example, both beams 1 and 4 maximize the function F_1. Since we already use the first processor, we map the remaining bandwidth to subsequent processors. We follow the same procedure as described above to select the processor with the lowest priority factor and allocate bandwidth chunks accordingly for the remaining processors. Hence, we select the second processor since it has the lowest priority factor than the third processor in this example. This processor operates at maximum efficiency with beams of 2 and 5. Finally, the third processor is selected and assigned to beam 3.



$\underset{\substack{y_l, z_i[l],\\ x_m[i][l], \forall_{i,l,m}}}{\texttt{minimize}} \sum_{l=1}^{L} w_l \frac{y_l}{F_l+1}$
y_1 w_1 F_1 [Hz]
y_2 w_2 F_2 [Hz]
y_L w_L F_3 [Hz]
$w_1 \leq w_2 \leq w_3$
Beam 1
Beam 2
Beam 3
Beam 4
Beam 5
$\underset{\substack{y_l, z_i[l],\\ x_m[i][l], \forall_{i,l,m}}}{\texttt{minimize}} \sum_{l=1}^{L} w_l \frac{y_l}{F_l+1} \equiv \underset{\substack{z_i[1],\\ x_m[i][1], \forall_{i,l,m}}}{\texttt{maximize}} F_1$
$y_1 = 1$ w_1
$y_2 = 0$ $y_3 = 0$
Beam 1
Beam 2
Beam 3
Beam 4
Beam 5
Beam 1
Beam 4
$\underset{\substack{y_l, z_i[l],\\ x_m[i][l], \forall_{i,l,m}}}{\texttt{minimize}} \sum_{l=1}^{L} w_l \frac{y_l}{F_l+1} \equiv \underset{\substack{z_i[1],\\ x_m[i][1], \forall_{i,l,m}}}{\texttt{maximize}} F_2$
$y_2 = 1$ w_2
$y_3 = 0$
Beam 1
Beam 2
Beam 3
Beam 4
Beam 5
Beam 2
Beam 5
$\underset{\substack{y_l, z_i[l],\\ x_m[i][l], \forall_{i,l,m}}}{\texttt{minimize}} \sum_{l=1}^{L} w_l \frac{y_l}{F_l+1} \equiv \underset{\substack{z_i[1],\\ x_m[i][1], \forall_{i,l,m}}}{\texttt{maximize}} F_3$
$y_3 = 1$ w_3
Beam 1
Beam 2
Beam 3
Beam 4
Beam 5
Beam 3

Fig. 6. Sequential optimization.

Note that in this sequential optimization, we take into account one processor at a time, then the minimization of $\sum_{l=1}^{L} w_l \frac{y_l}{F_l+1}$ in (20) is simplified to the maximization of F_l. Furthermore, $\mathcal{T}8$ and $\mathcal{T}10$ of (20) are not required for this sequential optimization because both constraints are only useful when considering multiple processors at once. Furthermore, we can discard $\mathcal{T}7$ and $\mathcal{T}11$ in this sequential optimization because both are equivalently represented by $\mathcal{T}9$. In the above context, for a single processor, (20) reduces to

$$\underset{\substack{z_l[l],\\ x_n[i][l], \forall_{i,n}}}{\texttt{maximize}} \sum_{i=1}^{N} z_l[i] \sum_{\substack{i \in \mathcal{G}_n,\\ \mathcal{G}_n \in \mathcal{Z}}} B_n \min\{\lceil B_n / B_{\text{ava}} \rceil, 1\}$$

$\texttt{s.t.}$

$$\begin{aligned}
&\mathcal{T}6 : \sum_{\substack{i \in \mathcal{G}_n,\\ \mathcal{G}_n \in \mathcal{Z}}} x_n[i][l] \leq 1 \quad \forall_n\\
&\mathcal{T}9 : x_n[i][l] \in \{0, 1\} \quad \forall_{n,i}\\
&\mathcal{T}12 : \sum_{i=1}^{N} \sum_{\substack{i \in \mathcal{G}_n,\\ \mathcal{G}_n \in \mathcal{Z}}} x_n[i][l] B_n \leq B_{\text{ava}}\\
&\mathcal{T}13 : \sum_{\substack{i \in \mathcal{G}_n,\\ \mathcal{G}_n \in \mathcal{Z}}} x_n[i][l] = z_l[i] \sum_{\substack{i \in \mathcal{G}_n,\\ \mathcal{G}_n \in \mathcal{Z}}} \min\{\lceil B_n / B_{\text{ava}} \rceil, 1\} \quad \forall_i\\
&\mathcal{T}15 : z_l[i] \in \{0, 1\} \quad \forall_i. \qquad (21)
\end{aligned}$$

Note that by observing the $\mathcal{T}13$ equality of both left and right parts of the equation, we equivalently rewrite F_l as

$$\begin{aligned}
F_l &= \sum_{i=1}^{N} \sum_{\substack{i \in \mathcal{G}_n,\\ \mathcal{G}_n \in \mathcal{G}}} x_n[i][l] B_n\\
&= \sum_{i=1}^{N} z_l[i] \sum_{\substack{i \in \mathcal{G}_n,\\ \mathcal{G}_n \in \mathcal{Z}}} B_n \min\{\lceil B_n / B_{\text{ava}} \rceil, 1\}. \qquad (22)
\end{aligned}$$

Algorithm 2: Sequential Optimization Method.

```
Input: $l \leftarrow 0$;
$\mathcal{I}_l \leftarrow \{\}$;
while $\left|\bigcup_{l=1}^{L} \mathcal{I}_l\right| < N$ do
    $l \leftarrow l + 1$
    Solve (21), for $i \notin \bigcup_{l=1}^{L} \mathcal{I}_l$
    Update $x_n[i][l] \leq 1 - z_l[i], \forall_{i,n}$
    $\mathcal{I}_l \leftarrow i, \forall_i$ if $z_l[i] == 1$
```

Then, we consider (22) as an objective function of (21). Equation (21) has less computational complexity than (20) because we reduce the search space of the optimization variables corresponding to bandwidth mapping, processor indicator, and CC from 2^{M+N+L}, 2^L, and 2^{L+N}, to $2^{|\mathcal{V}|+N}$, 0, and 2^N, respectively. Furthermore, (21) is an integer linear program that can be solved by a branch-and-bound method using MOSEK in CVX solver [46]. Algorithm 2 describes the sequential optimization method. First, the algorithm solves (21) for the first processor and records the beams corresponding to this processor into $\mathcal{I}_l$. Then, for the subsequent processor, the algorithm solves (21) for all beams not included in the set $\mathcal{I}_l$, $\forall_l$. Finally, it terminates when all the bandwidths of beams are mapped to the processor. This occurs when the total number selected beams are equal to N, i.e., $|\bigcup_{l=1}^{L} \mathcal{I}_l| = N$. The computational complexity of this algorithm is given by $\mathcal{O}(L2^N)$.

2) *Bin-Packing Method:* This method solves the bin-packing problem by packing different-sized items into a finite number of bins, each having a limited capacity [34]. We consider processors as bins and bandwidth chunks as items in this context. Let the set of bandwidth chunks per beam be denoted as $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_N\}$, where $\mathcal{B}_i$ is the set of bandwidth chunks for beam i. The normalized size of $\mathcal{B}_i$ is defined as W_i, which indicates that the overall bandwidth allocated to ith beam and normalized by B_{ava}. Hence, our goal is to pack or map W_i $\forall_i$ to the processors while taking into account the processor abstraction scenarios such as 1) the bandwidth chunks of $\mathcal{B}_i$ that correspond to beam i

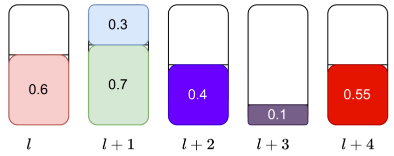


Fig. 7. Mapping bandwidth to processors using the next-fit algorithm.

must all be mapped to a single processor, 2) beams that share the same bandwidth must be mapped to different processors, and 3) to use as few processors as possible. In this manner, we consider the following bin-packing methods.

1) *Next-fit algorithm:* This activates only one bin/processor at a time. Furthermore, a new processor is activated when the item W_i does not fit in the current lth processor. In this case, the item W_i is mapped to the next $l+1$ processor if beam j in the lth processor and beam i belong to the same group or if the processor capacity is insufficient to support the W_i of beam i. As an example, consider $W_1 = 0.6$, $W_2 = 0.7$, $W_3 = 0.3$, $W_4 = 0.4$, $W_5 = 0.1$, and $W_6 = 0.55$ as well as the selected beam group:[3] $\mathcal{G}_7 =$\{Beam 1, Beam 2\}, $\mathcal{G}_{16} =$\{Beam 3, Beam 4\}, $\mathcal{G}_{22} =$\{Beam 1, Beam 2, Beam 6\}, $\mathcal{G}_{35} =$\{Beam 2, Beam 5, Beam 4\}, and $\mathcal{G}_{37} =$\{Beam 2, Beam 5, Beam 6\}. For this, assuming 5 processors, with each processor having the capacity of 1, the next fit does the following. The first step is to activate the processor l and determine whether the first item W_1 will fit. Since W_1 fits on the current processor, this algorithm places it on this processor. Next, the second item W_2 is selected to check if it fits on the processor l. In this case, the processor cannot accommodate the W_2 since the sum of W_1 and W_2 exceeds its capacity. The algorithm then closes the l processor and activates the $l+1$ processor to assign W_2. It then selects the next item W_3 to pack on the $l+1$ processor. Subsequently, the item W_4 is selected. Since beam 3 and beam 4 belong to the same group $\mathcal{G}_{16}$, W_4 cannot fit on the $l+1$ processor. Additionally, placing the item W_4 with W_2 and W_3 exceeds the processor's capacity. Hence, the algorithm activates the $l+2$ processor to pack W_4. Following that, W_5 and W_6 are placed on $l+3$ and $l+4$ processors, respectively. This is because beam 5 and beam 4 belong to the same group $\mathcal{G}_{35}$ and thus beam 5 cannot be placed on a $l+2$ processor. Additionally, since beam 6 and beam 5 belong to the same group $\mathcal{G}_{37}$, placing beam 6 in $l+3$ is not allowed. See Fig. 7 for this example. The computational complexity of this algorithm is given by $\mathcal{O}(L)$.

2) *First-fit algorithm:* In this algorithm, all processors are kept active and arranged in order. Then, it attempts to place each new item in the first processor available if it does not violate the orthogonality bandwidth and capacity of the processor. With the example above, the first-fit algorithm places W_1 on the l processor and W_2 on the $l+1$ processor. Then, W_3 is included in the l processor. Following this, W_4 is selected and placed on the $l+2$ processor since both l and $l+1$ have limited capacity. Furthermore, W_4 is not allowed to be with W_2 or W_3 because it belongs to the same group $\mathcal{G}_{35}$ and $\mathcal{G}_{16}$, respectively. Subsequently, W_5 is selected and included in the processor l. Finally, W_6 is assigned to $l+2$ processor because it cannot fit either in l and $l+1$ processor. In this case, we save two processors compared to the next-fit method. See Fig. 8 for this example. The computational complexity of this algorithm is given by $\mathcal{O}(L^2)$.

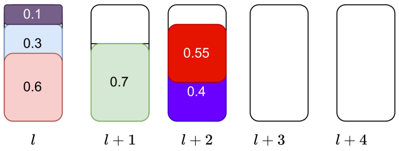


Fig. 8. Mapping bandwidth to processors using the first-fit algorithm.

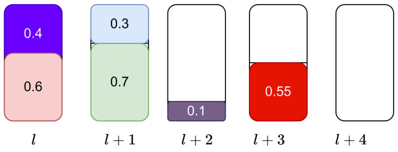


Fig. 9. Mapping bandwidth to processors using best-fit algorithm.

3) *Best-Fit algorithm:* Like the first-fit algorithm, all processors are in the order activated, but a new item is placed on the processor with the maximum load and does not violate the system's constraints described above. For the above example, W_1 and W_2 are packed into lth and $l+1$ processors, respectively. Then, the algorithm places W_3 to the processor with the maximum load that satisfies the system's constraints. For this, the $l+1$ processor has a higher load than the lth processor, and thus, W_3 is assigned to it. Subsequently, the item W_4 is placed on the lth processor because it has the maximum load and fulfills the system constraints. For item W_5, the algorithm assigned it to the $l+2$ processor because it cannot fit in the l or $l+1$ processor. Similarly, W_6 mapped into the $l+3$ processor. Note that beam 6 and beam 5 belong to the same group $\mathcal{G}_{37}$; thus, beam 6 cannot be placed in the $l+2$ processor. In this case, we save one processors compared to the next-fit method. See Fig. 9 for this example. The computational complexity of this algorithm is given by $\mathcal{O}(L^2)$. Hence, the remarks are as follows.

Remarks:

1) The next fit, first fit, and best fit have a polynomial complexity; thus, the respective solutions can be obtained from a computer in polynomial time.

[3] Note that beams in different groups use an orthogonal carrier frequency band while beams in the same group use the same carrier frequency band.

TABLE I
System Parameters

Parameter	Value
Satellite orbit	$-2.5°$E
Satellite altitude	8063 km
Number of beams (N)	20
Frequency band (f)	19 GHz (Ka)
Processor bandwidth (B_{ava})	500 MHz
Noise power density (N_0)	-204 dBW/Hz
Max. beam gain $(G_{\max})$	37.04 dBi
User antenna gain (G_R)	39.6 dBi
Power spectral density (S_{psd})	-78 dBW/Hz
Total available transmit power (P_{total})	158 W
Number of processors (L)	20
beam's demand threshold ϵ	1×10^{-6} b/s
Average rainfall rate $(\mathcal{P})$	1%

In contrast, sequential optimization has exponential complexity, which requires more time to obtain a solution from a computer than bin-packing methods. In this work, we assume a) the number of beams (N) equals the number of processors (L) and b) the bandwidth allocation per beam cannot exceed the operating processor bandwidth B_{ava}, which is indicated in constraints $\mathcal{T}1$ and $\mathcal{T}12$. Accordingly, the solution to the sequential optimization and the bin-packing methods is bounded by $1 \leq \mathcal{L} \leq L$.

IV. SIMULATION RESULTS

This section evaluates the performance of the demand-aware onboard payload processor management via numerical simulations. For this, we consider an MEO satellite operating at 8063 km above the Earth. Table I provides a summary of the key simulation parameters. In this simulation, we compare the proposed sequential optimization based on the branch-and-bound method and bin-packing algorithms, such as next fit, first fit, and best fit. For the sequential optimization method, we simulate the following two scenarios. The first is sequential optimization with CC, as explained in Section III-D1. The second scenario is when the sequential optimization uses the branch-and-bound method without CC (WoCC). In this case, the beam can be assigned to multiplier processors. However, beams that share a bandwidth chunk cannot belong to the same processor. The WoCC is obtained by removing the constraint $\mathcal{T}14$ of (21) and replacing the objective function with the upper-side equation of (22).

A. System Response to Uniform Demand

In this section, we demonstrate how the proposed algorithm behaves when there is a uniform demand. Specifically, we assume a demand of $D[i] = 20(\alpha + 2)\ \forall_i$ Mb/s with an integer value of $\alpha \in [1, 8]$. In addition, we average the simulation results from 400 Monte Carlo runs.

Fig. 10(a) shows the number of processors needed to accommodate uniform beam demands. The system requires fewer processors for low demands and many for high demands. Hence, as the demand increases, the number of active processors increases. This is because each beam needs more bandwidth for higher demands. Thus, more processors are required to handle the bandwidth of each beam. For example, five processors are adequate for 60 Mb/s, whereas 9 processors are sufficient for 100 Mb/s. Similarly, 12 processors are adequate for 160 Mb/s demand. However, as the demand increases to 200 Mb/s, the system can operate with 15 processors.

In addition, Fig. 10(a) compares the performance of the proposed schemes in utilizing the processors in response to the uniform beam demand. In this case, the branch-and-bound-WoCC uses fewer processors to handle each beam's bandwidth. This fewer-processor utilization results from branch-and-bound-WoCCs ability to map bandwidth chunks to multiple processors. However, this creates additional overhead for the system to recombine these chunks from multiple processors for transmission. The branch-and-bound-CC method requires fewer processors than next-fit, first-fit, and best-fit methods. For instance, at 100 Mb/s, it uses four processors. In contrast, next-fit, first-fit, and best-fit methods utilize nine, five, and five processors, respectively. For demand below 100 Mb/s and above 200 Mb/s, the first-fit, best-fit, and the branch-and-bound-CC methods have a similar performance. On the other hand, the next-fit method utilizes more processors than the other methods. This is because the algorithm only sees the current state of the processor and the current bandwidth. It has no prior knowledge of the remaining processors and bandwidth chunks. Hence, the next fit has limited flexibility in mapping the bandwidth to the processors. The branch and bound have flexibility in selecting beams to assign their corresponding bandwidth to each processor. In contrast, the best fit and first fit have processor selection freedom to map the current beam on the given processor. Generally, it is possible to implement the branch-and-bound method for a few beams. However, as the number of beams increases, the computational time for branch and bound may increase. Hence, the best-fit approach or the first-fit approach is best to employ for higher beams.

Fig. 10(b) depicts the amount of bandwidth handled in each processor when the beam demand is 200 Mb/s. The processors' load varies on average, with more loads in the lowest indexed processors than the highest ones. For example, processor 1 has more load than processor 16. This is because all methods map the bandwidth chunks sequentially to the processors based on their order. Hence, the lowest indexed processors are more likely to be checked by the algorithm, resulting in more bandwidth handling. However, the next-fit algorithm shows that processors from 1 to 14 each have similar loads. This is because once the algorithm knows a beam i is unfit for the lth processor, it closes the lth processor and begins on the $l + 1$ processor. Thus, checking each processor for every bandwidth chunk is less likely than the other methods. This results in slight load variations between the processors.

With the branch-and-bound-CC method, the performance is better than best-fit, first-fit, and next-fit methods, which fits as many beam bandwidths as possible in the order of the processors. However, it is less efficient than the

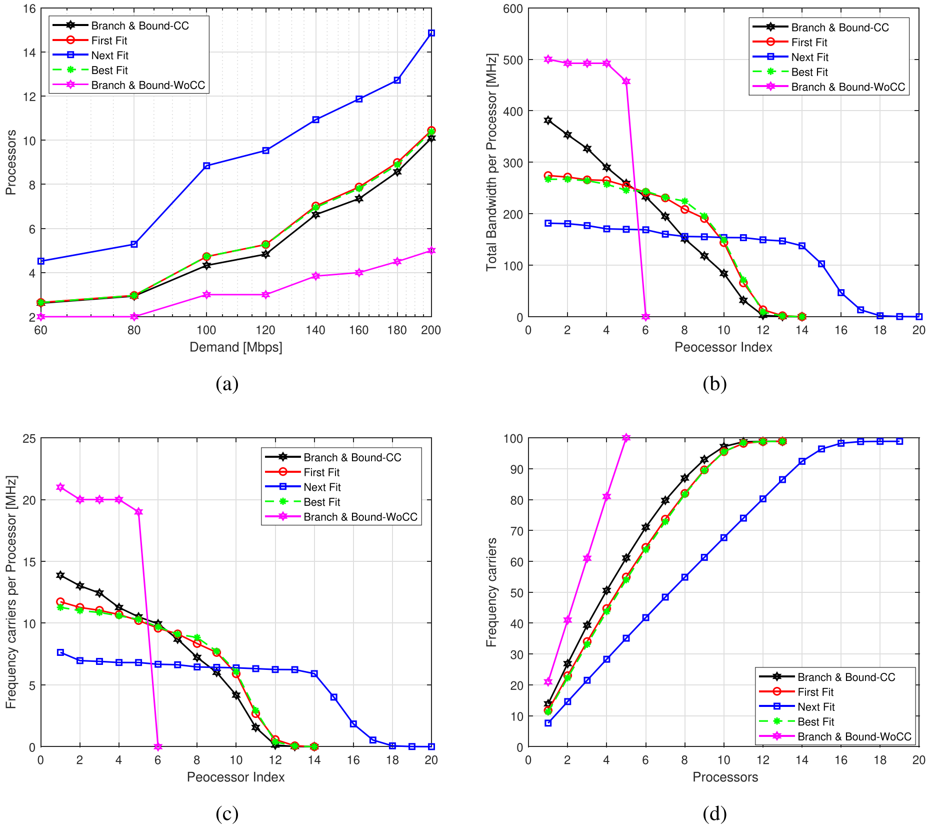


Fig. 10. Comparison of the proposed techniques for uniform demand distribution. (a) Processors used per demand. (b) Bandwidth per processor, for beam-demand = 200 Mb/s. (c) Number of carriers per processor, for beam-demand = 200 Mb/s. (d) Processors needed to process the number of carriers for beam-demand = 200 Mb/s.

branch-and-bound-WoCC method. Generally, having more bandwidth on a single processor results in free space on the remaining processor. Hence, we can use this for other applications. Similarly, Fig. 10(c) shows the number of carrier frequencies handled by each processor when each beam requests 200 Mb/s. For example, in the first processor, 8, 12, 12, 14, and 21 carriers are handled when the next-fit, best-fit, first-fit, branch-and-bound-CC, and branch-and-bound-WoCC methods are applied, respectively. Both branch-and-bound-WoCC and branch-and-bound-CC methods have better performance mapping the number of carriers to the processors than other methods.

Fig. 10(d) describes the processors required corresponding to frequency carriers of beams for 200 Mb/s. To satisfy all beams, we have 100 carriers handled by 11 processors when the methods branch-and-bound-CC, first fit, and best fit are applied. In contrast, the branch-and-bound-WoCC and next-fit methods handle the carriers in 4 and 20 processors, respectively. Hence, the branch-and-bound-WoCC method has better performance than other approaches. However, for practical implementation, it may require a complex signal processing technique to combine the carriers of each beam from multiple processors. While among the other methods, the branch-and-bound-CC method has better processor utilization. For example, given two and six processors, the branch-and-bound-CC method maps 27 and 71 carriers, respectively. In contrast, the first fit maps 23 and 65 carriers, respectively, whereas best fit maps 22 and 64 carriers. On the other hand, the next fit handles 15 and 42 carriers with 2 and 6 processors.

TABLE II
Mapping of Carriers to Processors in Percent for the Demand of 200 Mb/s

In-use Processors	Branch-and-bound-WoCC	Branch-and-bound-CC	First fit	Best fit	Next fit
1	21.0000	14.0321	11.8645	11.4067	7.7192
2	41.0000	27.2017	23.2662	22.5479	14.7655
3	61.0000	39.7845	34.4302	33.5399	21.7462
4	81.0000	51.1761	45.2476	44.2587	28.6383
5	100.0000	61.8064	55.5769	54.6841	35.5278
6		71.8701	65.2739	64.4873	42.2758
7		80.6718	74.5081	73.6962	48.9833
8		87.9812	82.9531	82.6344	55.5112
9		94.0639	90.6672	90.4497	62.0036
10		98.2852	96.6361	96.5906	68.4607
11		99.8584	99.3424	99.5498	74.8495
12		99.9874	99.9267	99.9494	81.1675
13		100.0000	100.0000	100.0000	87.4753
14					93.4569
15					97.5138
16					99.3778
17					99.9216
18					99.9899
19					100.0000
20					

Generally, all methods perform better on processor utilization than next fit. For this, see Table II frequency carriers versus processors in percent for 200 Mb/s. For example, with eight processors, branch-and-bound-CC, first-fit, and best-fit methods can handle 88%, 83%, and 83% of carriers, respectively. However, the next-fit method can only manage 56% of carries. In contrast, branch-and-bound-WoCC can efficiently operate in 100% cases with five processors.

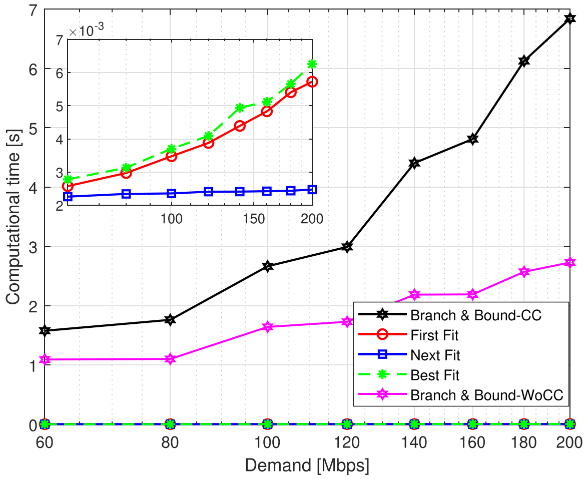


Fig. 11. Computational time of branch-and-bound and bin-packing methods for different demands.

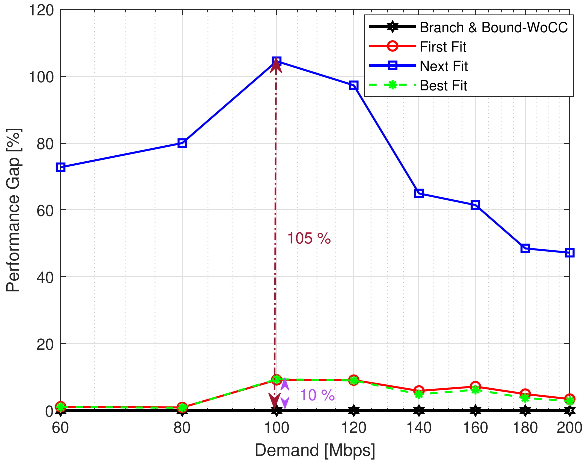


Fig. 12. Bin-packing performance gap compared to branch-and-bound-CC for different demands.

Fig. 11 shows the computational time for all schemes. We observe that the branch-and-bound method takes more time than bin-packing methods. For example, for a 100-Mb/s demand, the computational time of the branch-and-bound-CC and branch-and-bound-WoCC is 2.7 s and 1.65 s, respectively. In contrast, the computational time for next fit, first fit, and best fit at 100 Mb/s is 2.4 ms, 3.5 ms, and 3.7 ms, respectively. Typically, the bin-packing method uses a heuristic approach to map bandwidth chunks to processors. This requires less computational time compared to the analytical optimization approach of branch and bound.

Generally, the proposed algorithm takes less time to compute, permitting us to control the payload processors in real time. Moreover, if timing becomes a constraint, the proposed method can be used as a basis for an adaptive algorithm to control the number of active processors. This is due to a slow and continuous variation in traffic demand as well as communication channels on a satellite-terrestrial link. Hence, the algorithm may not be recomputed each time for a slight change in the parameters. However, this is beyond the scope of this work.

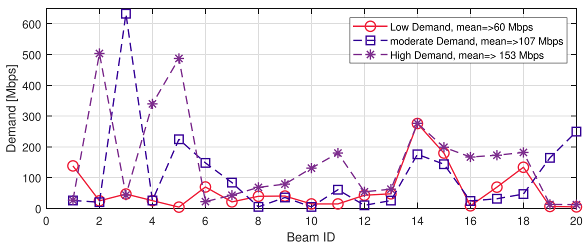


Fig. 13. Heterogeneous demand.

Fig. 12 shows the performance gap[4] of the bin-packing method while considering the branch-and-bound-CC method as a baseline. The performance gap of the next-fit method is significantly higher than that of the first-fit and best-fit methods. For instance, at 100 Mb/s, the performance gap of next fit is 105%, whereas the performance gap for first fit and best fit is 10%. Generally, since the next-fit method has a lower computational time, it is suitable for time-constrained scenarios at the expense of requiring more processors. If the time constraint is not critical, it is possible to implement the branch-and-bound-CC method to utilize fewer processors while a tradeoff can be achieved between the number of active processors and computation time using the first-fit and best-fit methods.

B. Heterogeneous Demand

In this section, we study the performance of the proposed methods for heterogeneous demand. The traffic demand can be modeled as a Poisson distribution which follows an exponential distribution [47]. Hence, we generate each beam demand using an exponential distribution. The cumulative distribution function of this distribution is provided as

$$F(D[i]) = 1 - \exp\left(-\frac{D[i]}{\beta}\right). \tag{24}$$

Here, we assume β and $D[i]$ are the mean traffic demand of the system and random variable indicating the demand of beam i, respectively. Then, we obtain demand $D[i]$ from $F(D[i])$ using the inverse method [48]. For this, we generate a uniform random number $F(D[i]) = \chi[i]$ in the interval (0,1). Hence, $D[i]$ is obtained as follows:

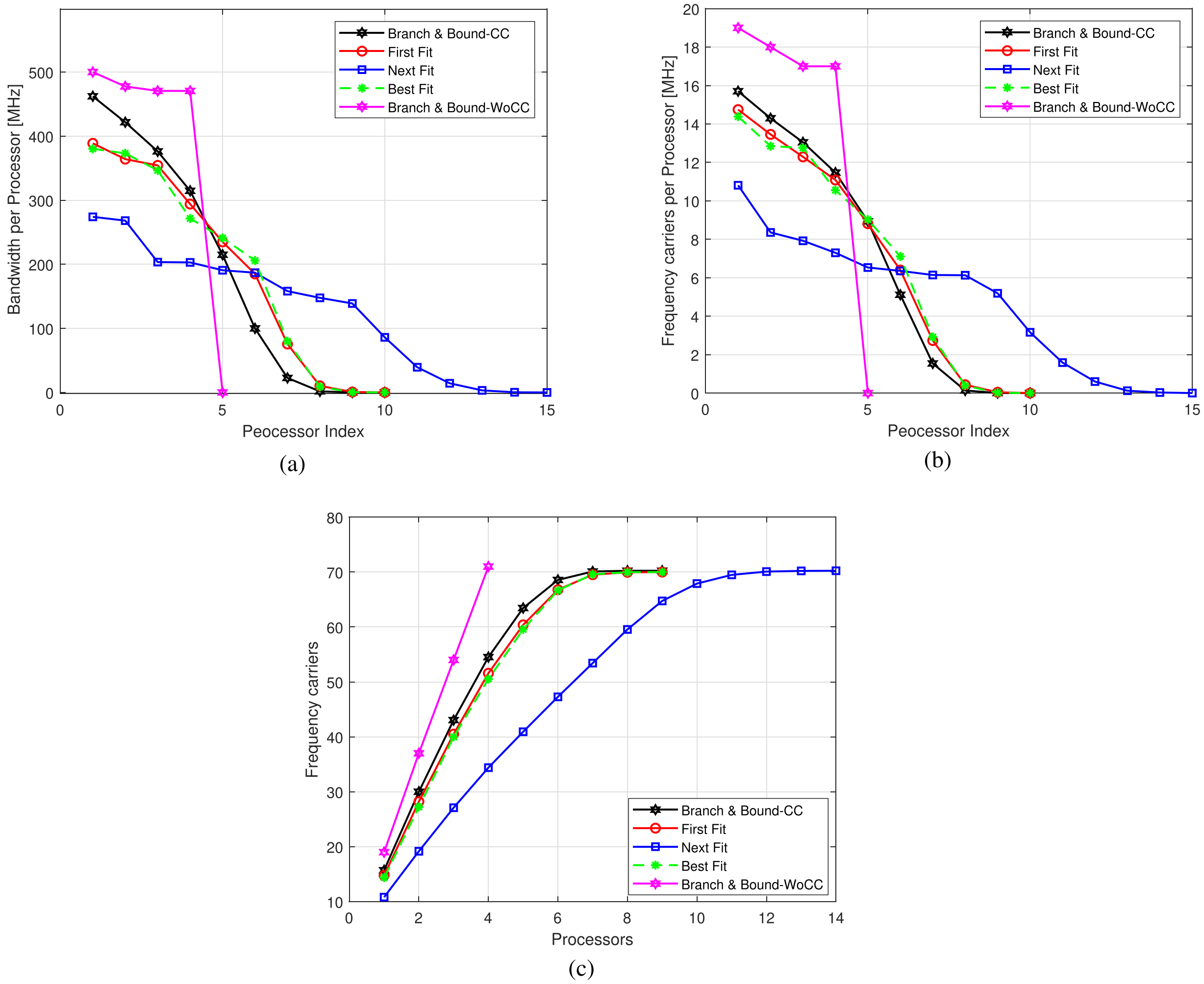
$$D[i] = -\beta \log(1 - \chi[i]). \tag{25}$$

Accordingly, we generate three heterogeneous demand[5] distributions based on the (25), which are called low, moderate, and high demand, as shown in Fig. 13.

[4] The performance gap in percent is given by

$$\text{Performance Gap } = 100\frac{\text{Proposed method} - \text{branch-and-bound-CC}}{\text{branch-and-bound-CC}} \tag{23}$$

[5] Note that we use the mean demand to represent each demand distribution. In the case of low demand, the mean of a demand distribution is less than the mean of other demand distributions. In contrast, high demand indicates that the mean of a demand distribution is higher than that of other mean demand distributions. On the other hand, moderate demand implies that the mean demand distribution lies between low and high demand.



(a)

(b)

(c)

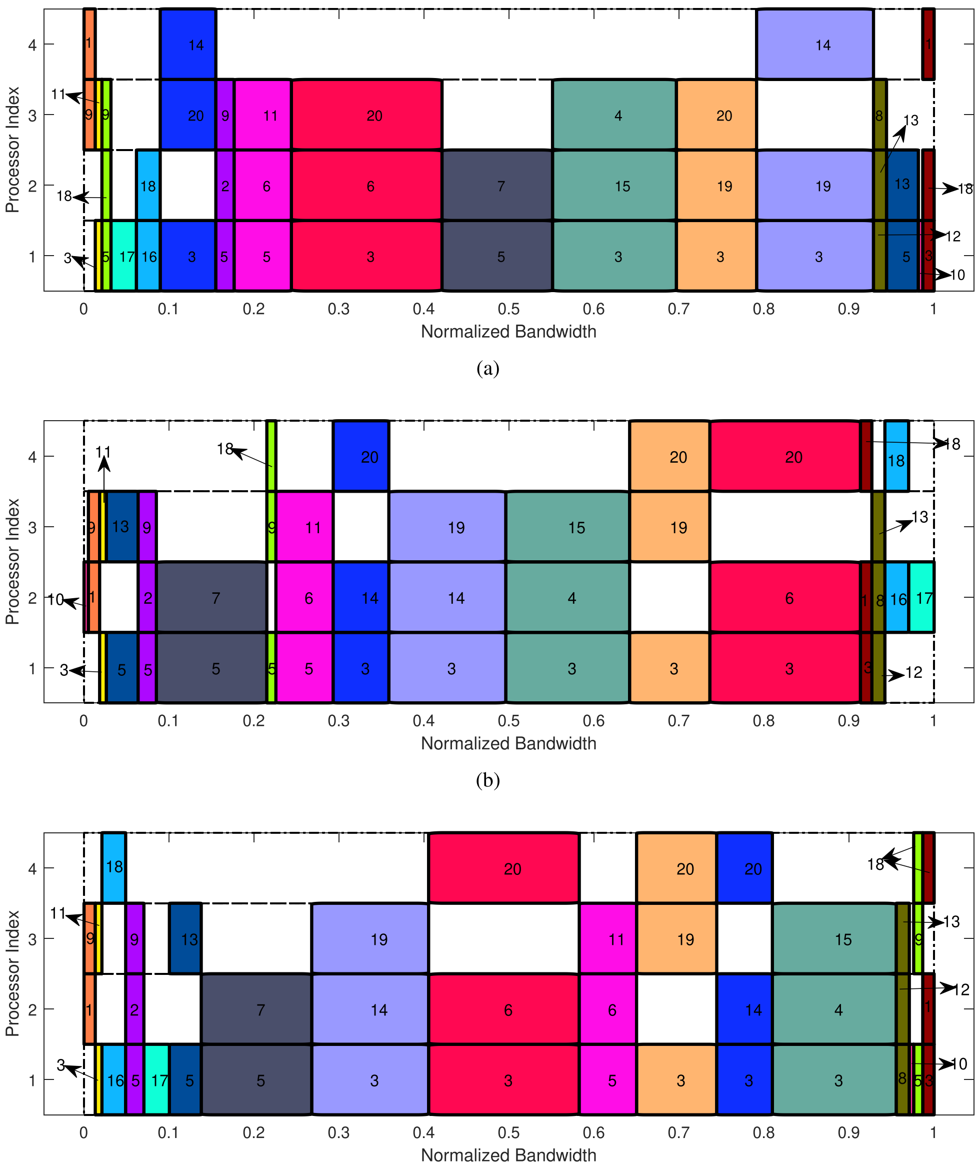
Fig. 14. Comparison of the proposed techniques for heterogeneous demand distribution. (a) Bandwidth per processor, for the high-demand scenario. (b) Number of carriers per processor, for the high-demand scenario. (c) Processors needed to process the number of carriers for the high-demand scenario.

TABLE III
Processors Used per Demand

		Low Demand	Moderate Demand	High Demand
Processors	Branch-and-bound-WoCC	2	3	5
	Branch-and-bound-CC	3	4	6
	First Fit	3	5	7
	Best Fit	3	5	7
	Next Fit	4	7	10

Table III shows the processor usage in case of low-, moderate-, and high-demand distributions. Each distribution is represented in the table by its mean value. We observe that the system uses few processors when beams have low demand while more when moderate and high demands occur. However, the moderate-demand distribution requires fewer processors than a high-demand distribution. The branch-and-bound-CC method has better processor utilization than first-fit, next-fit, and best-fit methods for moderate- and high-demand distribution. For instance, the branch-and-bound-CC uses four processors at moderate demand, whereas the first fit, the best fit, and the next fit, respectively, utilize five, five, and seven processors. However, all the proposed methods except next fit show the same performance in the case of low demand. As expected, the branch-and-bound-WoCC has better processor utilization than the other methods. However, it does not support CC, so it cannot assign all bandwidth chunks of a beam to one processor. Hence, it needs an additional signal processing technique to recombine the bandwidth chunks of a beam from multiple processors for transmission. This problem does not occur in other methods that handle a beam's bandwidth chunks in a single processor.

Fig. 14(a) depicts the overall bandwidth of signals carried out in each processor when the demand is high. The processors with the lowest index values handle most bandwidth chunks instead of those with the highest index values. Similarly, the frequency carriers managed by each processor are shown in Fig. 14(b). The first processor executes most frequency carriers, while other processors handle the remaining ones. This is because the algorithm runs sequentially to use each processor. Hence, it utilizes the first processor before checking the next processor.



(a)

(b)

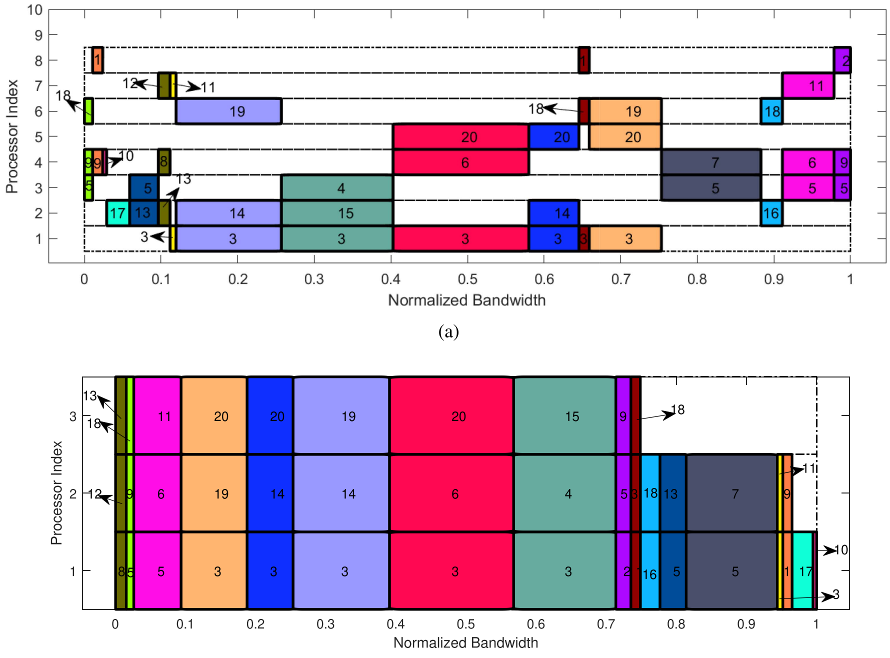
(c)

Fig. 15. Arrangement of bandwidth chunks on the processors. (a) Branch-and-bound-CC. (b) First fit. (c) Best fit.

Additionally, we can see from Fig. 14(c) that fewer processors handle the majority of the carriers. For example, using only six processors, the system can deal with 94%–97% of the carriers with the branch-and-bound-CC, first-fit, and best-fit methods. Hence, we can save up to three processors.

C. Arrangement of Bandwidth Chunks on the Processors

The following section demonstrates the distribution of the bandwidth chunks corresponding to each beam among the processors. Here, we consider a moderate demand scenario, and Fig. 15 shows the arrangement of the bandwidth chunks. Specifically, Fig. 15(a)–(c) depicts the arrangement of the normalized bandwidth chunks of each beam on the processors for branch-and-bound-CC, first-fit, and best-fit methods, respectively. Each figure has a beam number, and the color indicates the bandwidth chunk per beam. Furthermore, two or more beams share the same bandwidth chunk represented by the same color. For example, beams 3, 6, and 20 are represented by red color since they share the same bandwidth chunk. In addition, Fig. 16(a) and (b) provides the arrangement of the normalized bandwidth chunks on the processors when the system employs next-fit and branch-and-bound-WoCC methods, respectively. We observe that branch-and-bound-WoCC has a better bandwidth arrangement compared to other methods. This is because the method supports distributing the bandwidth chunks of any beam to multiple processors. For instance, beam 3, 6, and 20 in Fig. 16(b) utilizes multiple processors. Therefore, the flexibility to assign bandwidth chunks to each processor increases. Hence, it utilizes fewer processors than other methods. However, it requires more signal processing techniques to recombine the respective bandwidth chunks



(a)

(b)

Fig. 16. Arrangement of bandwidth chunks on the processors. (a) Next fit. (b) Branch-and-bound-WoCC.

of beams. Unlike branch-and-bound-WoCC, none of the remaining methods require additional signal processing since the bandwidth chunks of a beam are processed only in one processor. For example, beam 3 uses only the first processor.

The branch-and-bound-CC on processor one, the best fit on processor two, and the branch-and-bound-WoCC on processor one, shown in Fig. 15(a), (c), and Fig. 16(b), respectively, have more bandwidth chunks compared to the first-fit and next-fit methods. Consequently, the remaining processors will have more space for managing other applications. Regarding the number of processors utilized, the branch-and-bound-CC, first fit, and best fit show similar performance. On the other hand, the next-fit method shown in Fig. 16(a) utilizes more processors than other methods. Furthermore, this creates a significant amount of unused bandwidth in each processor. In contrast, branch-and-bound-WoCC uses fewer processors than the other methods.

V. CONCLUSION

In this article, we propose an approach in this article to manage onboard processors for high-throughput NGSO satellite systems. For this, we develop algorithms that can flexibly control the number of utilized payload processors in response to the system and user requirements. Accordingly, we formulated an optimization problem to minimize the number of operating processors under beam abstraction, demand satisfaction, and processor abstraction constraints.

The optimization problem is nonconvex, and we solve it in two steps. First, we design the bandwidth allocation strategy. Subsequently, we determine the exact number of processors required to accommodate this bandwidth allocation. In this context, we propose a sequential optimization-based branch-and-bound method and bin-packing method using next fit, first fit, and best fit. Consequently, we evaluate the performance of each of the proposed methods. As a result, the branch-and-bound, best-fit, and first-fit algorithms provide the best results in flexibly managing payload processors.

REFERENCES

[1] B. Al Homssi et al., "Next generation mega satellite networks for access equality: Opportunities, challenges, and performance," *IEEE Commun. Mag.*, vol. 60, no. 4, pp. 18–24, Apr. 2022.

[2] O. Kodheli et al., "Satellite communications in the new space era: A survey and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 70–109, Jan.–Mar. 2021.

[3] D. Peng, D. He, Y. Li, and Z. Wang, "Integrating terrestrial and satellite multibeam systems toward 6G: Techniques and challenges for interference mitigation," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 24–31, Feb. 2022.

[4] H. Al-Hraishawi, H. Chougrani, S. Kisseleff, E. Lagunas, and S. Chatzinotas, "A survey on non-geostationary satellite systems: The communication perspective," *IEEE Commun. Surveys Tut.*, vol. 25, no. 1, pp. 101–132, Firstquarter 2023, doi: 10.1109/COMST.2022.3197695.

[5] T. M. Braun and W. R. Braun, "Processing payload and flexible payload," in *Proc. Satell. Commun. Payload Syst.*, 2021, pp. 270–296.

[6] A. Y. Garcia, J. M. R. Bejarano, I. Jimenez, and J. Prat, "Future opportunities for next generation OBP enhanced satellite payloads," in *Proc. 31st AIAA Int. Commun. Satell. Syst. Conf.*, 2013, Art. no. 5656. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2013-5656

[7] J. P. Choi and V. W. S. Chan, "Optimum power and beam allocation based on traffic demands and channel conditions over satellite downlinks," *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 2983–2993, Nov. 2005.

[8] X. P. H. Wang, A. Liu, and J. Li, “Optimization of power allocation for a multibeam satellite communication system with interbeam interference,” *J. Appl. Math.*, vol. 2014, pp. 1–8, Jan. 2014.

[9] A. I. Aravanis et al., “Power allocation in multibeam satellite systems: A two-stage multi-objective optimization,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3171–3182, Jun. 2015.

[10] C. N. Efrem and A. D. Panagopoulos, “Dynamic energy-efficient power allocation in multibeam satellite systems,” *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 228–231, Feb. 2020.

[11] J. J. G. Luis, M. Guerster, I. del Portillo, E. Crawley, and B. Cameron, “Deep reinforcement learning for continuous power allocation in flexible high throughput satellites,” in *Proc. IEEE Cogn. Commun. Aerosp. Appl. Workshop*, 2019, pp. 1–4.

[12] S. Liu, Y. Fan, Y. Hu, D. Wang, L. Liu, and L. Gao, “AG-DPA: Assignment game–based dynamic power allocation in multibeam satellite systems,” *Int. J. Satell. Commun. Netw.*, vol. 38, no. 1, pp. 74–83, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1310

[13] F. Li, K. Lam, X. Liu, J. Wang, K. Zhao, and L. Wang, “Joint pricing and power allocation for multibeam satellite systems with dynamic game model,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2398–2408, Mar. 2018.

[14] P. Zhang, X. Wang, Z. Ma, S. Liu, and J. Song, “An online power allocation algorithm based on deep reinforcement learning in multibeam satellite systems,” *Int. J. Satell. Commun. Netw.*, vol. 38, no. 5, pp. 450–461, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1352

[15] U. Park, H. W. Kim, D. S. Oh, and B. J. Ku, “Flexible bandwidth allocation scheme based on traffic demands and channel conditions for multi-beam satellite systems,” in *Proc. IEEE Veh. Technol. Conf.*, 2012, pp. 1–5.

[16] X. Hu et al., “Multi-agent deep reinforcement learning-based flexible satellite payload for mobile terminals,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9849–9865, Sep. 2020.

[17] G. Cocco, T. de Cola, M. Angelone, Z. Katona, and S. Erl, “Radio resource management optimization of flexible satellite payloads for DVB-S2 systems,” *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 266–280, Jun. 2018.

[18] F. G. Ortiz-Gomez, D. Tarchi, R. Martínez, A. Vanelli-Coralli, M. A. Salas-Natera, and S. Landeros-Ayala, “Cooperative multi-agent deep reinforcement learning for resource management in full flexible VHTS systems,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 1, pp. 335–349, Mar. 2022.

[19] P. V. R. Ferreira et al., “Multiobjective reinforcement learning for cognitive satellite communications using deep neural network ensembles,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 1030–1041, May 2018.

[20] P. V. R. Ferreira et al., “Reinforcement learning for satellite communications: From LEO to deep space operations,” *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 70–75, May 2019.

[21] F. G. Ortiz-Gomez, D. Tarchi, R. Martínez, A. Vanelli-Coralli, M. A. Salas-Natera, and S. Landeros-Ayala, “Convolutional neural networks for flexible payload management in VHTS systems,” *IEEE Syst. J.*, vol. 15, no. 3, pp. 4675–4686, Sep. 2021.

[22] T. S. Abdu, E. Lagunas, S. Kisseleff, and S. Chatzinotas, “Carrier and power assignment for flexible broadband GEO satellite communications system,” in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun.*, 2020, pp. 1–7.

[23] T. S. Abdu, S. Kisseleff, E. Lagunas, and S. Chatzinotas, “Limits of smart radio resource assignment in GEO satellite communications,” in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–7.

[24] T. S. Abdu, S. Kisseleff, E. Lagunas, and S. Chatzinotas, “Flexible resource optimization for GEO multibeam satellite communication system,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7888–7902, Dec. 2021.

[25] K. An, M. Lin, J. Ouyang, and W.-P. Zhu, “Secure transmission in cognitive satellite terrestrial networks,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 3025–3037, Nov. 2016.

[26] Z. Lin, M. Lin, J.-B. Wang, T. de Cola, and J. Wang, “Joint beamforming and power allocation for satellite-terrestrial integrated networks with non-orthogonal multiple access,” *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 657–670, Jun. 2019.

[27] Z. Lin, M. Lin, T. de Cola, J.-B. Wang, W.-P. Zhu, and J. Cheng, “Supporting IoT with rate-splitting multiple access in satellite and aerial-integrated networks,” *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11123–11134, Jul. 2021.

[28] Z. Yin et al., “UAV-assisted physical layer security in multibeam satellite-enabled vehicle communications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2739–2751, Mar. 2022.

[29] Z. Yin, N. Cheng, T. H. Luan, and P. Wang, “Physical layer security in cybertwin-enabled integrated satellite-terrestrial vehicle networks,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4561–4572, May 2022.

[30] Z. Yin, N. Cheng, T. H. Luan, Y. Hui, and W. Wang, “Green interference based symbiotic security in integrated satellite-terrestrial communications,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 9962–9973, Nov. 2022.

[31] K. Kaneko, H. Nishiyama, N. Kato, A. Miura, and M. Toyoshima, “Construction of a flexibility analysis model for flexible high-throughput satellite communication systems with a digital channelizer,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2097–2107, Mar. 2018.

[32] Y. Kawamoto, T. Kamei, M. Takahashi, N. Kato, A. Miura, and M. Toyoshima, “Flexible resource allocation with inter-beam interference in satellite communication systems with a digital channelizer,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 2934–2945, May 2020.

[33] A. I. Perez-Neira, M. A. Vazquez, M. B. Shankar, S. Maleki, and S. Chatzinotas, “Signal processing for high-throughput satellites: Challenges in new interference-limited scenarios,” *IEEE Signal Process. Mag.*, vol. 36, no. 4, pp. 112–131, Jul. 2019.

[34] B. Korte and J. Vygen, *Bin-Packing*. Berlin, Germany: Springer, 2000, pp. 407–422, doi: 10.1007/978-3-662-21708-5_18.

[35] E. Amyotte et al., “A summary of recent developments in satellite antennas at MDA,” in *Proc. 5th Eur. Conf. Antennas Propag.*, 2011, pp. 3203–3207.

[36] “Introducing O3b mPOWER,” Mar. 8, 2018. Accessed: 7, 2022. [Online]. Available: https://www.ses.com/blog/introducing-o3b-mpower

[37] A. Mallet, A. Anakabe, J. Sombrin, and R. Rodriguez, “Multiport-amplifier-based architecture versus classical architecture for space telecommunication payloads,” *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 12, pp. 4353–4361, Dec. 2006.

[38] D. Christopoulos, S. Chatzinotas, G. Zheng, J. Grotz, and B. Ottersten, “Linear and nonlinear techniques for multibeam joint processing in satellite communications,” *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, pp. 1–13, 2012.

[39] *The RF Link*. Hoboken, NJ, USA: Wiley, 2008, ch. 4, pp. 51–76. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470754443.ch4

[40] C. Caini, G. Corazza, G. Falciasecca, M. Ruggieri, and F. Vatalaro, “A spectrum- and power-efficient EHF mobile satellite system to be integrated with terrestrial cellular systems,” *IEEE J. Sel. Areas Commun.*, vol. 10, no. 8, pp. 1315–1325, Oct. 1992.

[41] M. A. Diaz, N. Courville, C. Mosquera, G. Liva, and G. E. Corazza, “Non-linear interference mitigation for broadband multimedia satellite systems,” in *Proc. Int. Workshop Satell. Space Commun.*, 2007, pp. 61–65.

[42] *Propagation Data and Prediction Methods Required for the Design of Earth-Space Telecommunication Systems*, Rec. ITU-R P. 618-13, International Telecommunications Union, Geneva, Switzerland, 2017.

[43] *Rain Height Model for Prediction Methods*, Rec. ITU-RTU-R P. 839-4, International Telecommunications Union, Geneva, Switzerland, 2011.

[44] *Characteristics of Precipitation for Propagation Modelling*, Rec. ITU-R P. 837-7, International Telecommunications Union, Geneva, Switzerland, 2017.

[45] *Specific Attenuation Model for Rain for Use in Prediction Methods*, Rec. ITU-R P. 838-3, International Telecommunications Union, Geneva, Switzerland, 2005.

[46] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[47] V. Frost and B. Melamed, “Traffic modeling for telecommunications networks,” *IEEE Commun. Mag.*, vol. 32, no. 3, pp. 70–81, Mar. 1994.

[48] L. Devroye, “Chapter 4: Nonuniform random variate generation,” in *Simulation* (Handbooks in Operations Research and Management Science Series), S. G. Henderson and B. L. Nelson, Eds., vol. 13. New York, NY, USA: Elsevier, 2006, pp. 83–121. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927050706130042

Tedros Salih Abdu (Member, IEEE) received the joint M.Sc. degree in electrical engineering from Pan African University and the Jomo Kenyatta University of Agriculture and Technology, Juja, Kenya, in 2017, and the Ph.D. degree in computer science from the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg, in 2022.

From 2017 to 2019, he was a Lecturer with the Kombolcha Institute of Technology, Wollo University, and, from 2019 to 2022, he was a Doctoral Researcher with the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, where he is currently a Research Associate. His research interest includes wireless communication technology, focusing on channel modeling, channel estimation, precoding, and resource optimization.

Steven Kisseleff (Senior Member, IEEE) received the M.Sc. degree in information technology from the Technical University of Kaiserslautern, Kaiserslautern, Germany, in 2011, and the Ph.D. degree in electrical engineering from the Friedrich-Alexander University of Erlangen-Nürnberg (FAU), Erlangen, Germany, in 2017.

He was a Research and Teaching Assistant with the Institute for Digital Communications, FAU, from October 2011 to July 2018. In 2012 and 2013, he was a Visiting Researcher with the State University of New York at Buffalo, USA, and with Broadband Wireless Networking Lab, Georgia Institute of Technology, Atlanta, USA, respectively. In 2018, he joined the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg, where he is currently a Research Scientist. His main research interests include satellite communications, wireless power transfer, symbol-level precoding, and reconfigurable intelligent surfaces.

Eva Lagunas (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Polytechnic University of Catalonia (UPC), Barcelona, Spain, in 2010 and 2014, respectively.

She was a Research Assistant with the Department of Signal Theory and Communications, UPC, from 2009 to 2013. In Summer 2009, she was a Guest Research Assistant with the Department of Information Engineering, Pisa, Italy. From November 2011 to May 2012, she held a visiting research appointment with the Center for Advanced Communications, Villanova University, Villanova, PA, USA. In 2014, she joined the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg, where she is currently a Research Scientist. Her research interests include radio resource management and general wireless networks optimization.

Joël Grotz (Senior Member, IEEE) received the degree in electrical engineering from the University of Karlsruhe and the Grenoble Institute of Technology in 1999, and the Ph.D. degree in telecommunications jointly from the University of Luxembourg and KTH, Stockholm, in 2008. He worked with SES, Betzdorf, Luxembourg, on the development of satellite broadband communication system design for GEO and MEO high-throughput satellite systems on different ground segment and space segment topics and system optimization aspects.

He worked with the Technical Labs at ST Engineering iDirect (former Newtec Cy at Sint-Niklaas in Belgium) on topics of system design and signal processing in satellite modems. He is currently working as a Senior Manager at SES on the development of a dynamic resource management system for novel flexible satellite systems, including SES-17 and O3b mPOWER as well as future satellite systems under planning.

Symeon Chatzinotas (Fellow, IEEE) received the M.Eng. degree in telecommunications from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Surrey, Surrey, U.K., in 2006 and 2009, respectively.

He is currently a Full Professor/Chief Scientist I and the Head of the SIGCOM Research Group, SnT, University of Luxembourg, Luxembourg, Luxembourg. Previously, he was a Visiting Professor with the University of Parma, Italy, and he was involved in numerous research and development projects for the National Center for Scientific Research Demokritos, the Center of Research and Technology Hellas, and the Center of Communication Systems Research, University of Surrey. He has (co)authored more than 500 technical articles in refereed international journals, conferences, and scientific books.

Dr. Chatzinotas was a corecipient of the 2014 IEEE Distinguished Contributions to Satellite Communications Award, the CROWNCOM 2015, the 2018 EURASIC JWCN, and the 2022 WCNC Paper Award. He is currently on the editorial board for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY, and the *International Journal of Satellite Communications and Networking*.

Björn Ottersten (Fellow, IEEE) received the M.S. degree in electrical engineering and applied physics from Linköping University, Linköping, Sweden, in 1986, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1990.

He has held research positions with the Department of Electrical Engineering, Linköping University, the Information Systems Laboratory, Stanford University, the Katholieke Universiteit Leuven, Leuven, Belgium, and the University of Luxembourg, Luxembourg. From 1996 to 1997, he was the Director of Research with ArrayComm, Inc., a start-up in San Jose, CA, USA, based on his patented technology. In 1991, he was a Professor of signal processing with the Royal Institute of Technology (KTH), Stockholm, Sweden. He is the Head of the Department for Signals, Sensors, and Systems, KTH, and the Dean of the School of Electrical Engineering, KTH. He is currently the Director of the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg.

Dr. Ottersten is a recipient of the IEEE Signal Processing Society Technical Achievement Award, the EURASIP Group Technical Achievement Award, and the European Research Council advanced research grant twice. He has coauthored journal papers that received the IEEE Signal Processing Society Best Paper Award in 1993, 2001, 2006, 2013, and 2019, and eight IEEE conference papers best paper awards. He has been a board member of the IEEE Signal Processing Society, the Swedish Research Council, and currently serves on the boards of EURASIP and the Swedish Foundation for Strategic Research. He was the Editor-in-Chief of *EURASIP Signal Processing* and served on the editorial boards of IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE SIGNAL PROCESSING MAGAZINE, IEEE OPEN JOURNAL FOR SIGNAL PROCESSING, *EURASIP Journal of Advances in Signal Processing*, and *Foundations and Trends in Signal Processing*. He is a Fellow of EURASIP.