A Deep Learning Based Acceleration of Complex Satellite Resource Management Problem

Tedros Salih Abdu*, Steven Kisseleff*, Lei Lei†, Eva Lagunas*, Joel Grotz‡ and Symeon Chatzinotas*
*Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg
†School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an, China
‡ SES Engineering, Betzdorf, Luxembourg
Email: *tedros-salih.abdu@uni.lu

***Abstract*—Demand-based algorithms have been widely studied in the satellite community, where the satellite's radio resources are allocated according to the on-ground users' demands. Hence, we can accommodate the increasing demand while efficiently utilizing satellite resources. However, these algorithms have high computational time because they are required to optimize more parameters, which hinders the practical implementation of the algorithms. In this paper, we propose a methodology to alleviate the computational complexity of demand-aware bandwidth and power allocation algorithm by combining conventional optimization and deep learning (DL). Hence, conventional optimization allocates the radio resources, while DL accelerates the computation. The simulation result shows that the proposed approach has lower computational time while efficiently utilizing the resource of the satellite.**

***Index Terms*—Bandwidth allocation, deep learning, power allocation.**

I. Introduction

Satellite resources, such as power and bandwidth, are limited and must be efficiently managed to cope with the increasing demand. Thanks to reconfigurable digital payload technologies, it is now possible to control satellite resources in response to user demands by adjusting the payload transponders' frequency, bandwidth, and power. Hence, satellite resources can be allocated unevenly according to user demand, where a small amount is allocated to low-demand users and a large portion to high-demand users [1]. However, the reconfigurable payload must be aided by an efficient algorithm to achieve this demand-based allocation. Several studies have been conducted to develop demand-aware algorithms that can be classified into three types according to their complexity and optimally:

1) Analytical optimization: The algorithm provides an optimal or nearly optimal solution using this method. In this context, the power allocation algorithm in [2] has been developed to satisfy the requested demand. Similarly, in [3], joint power and bandwidth optimization has been considered to match the demand per user. However, the proposed methods require a large number of optimization parameters and are therefore very time-consuming. Hence, the proposed resource allocation algorithms may not suit real-time systems.
2) Meta-heuristic optimization: This method may not guarantee the optimality, but it is well-suited for problems involving non-linear, multi-objective, and NP-hard problems. In this context, power allocation in [4] has been considered to match the demand per user. Furthermore, in [5], joint bandwidth and power allocation has been proposed for demand matching.
3) Machine learning: Algorithms in this category interact with the environment or data to predict or decide possible solutions. Like meta-heuristic methods, machine learning algorithms may not guarantee optimal solutions. However, machine learning methods require considerably less computation time than meta-heuristic methods [6]. In this context, a power allocation in [7] and bandwidth allocation in [8] based on reinforcement learning has been investigated to meet the requested demand. In [9], and [10], a multi-objective optimization approach has been used with supervised learning and reinforcement learning, respectively, to utilize the satellite resources.

In this paper, we examine the synergy between analytical optimization and DL to design a learning-assisted demand-based resource allocation algorithm. Hence, we can analytically optimize the bandwidth allocation while dramatically reducing the computational complexity via DL. Although DL for scheduling in [11] has been already investigated, the bandwidth allocation using DL has not been explored yet. Additionally, DL with genetic algorithm is used in [12] to allocate radio resources. However, in this paper, we make use of DL to narrow down the huge search space of the radio resource allocation problem, thus accelerating the convergence of optimization-based solutions. The contributions of this paper can be summarized as follows.

1) Firstly, we formulate a demand-aware bandwidth and power optimization problem for a broadband Geostationary(GEO) satellite system. For this, we design a utility function to minimize the demand-offered data mismatch and the overall bandwidth consumption.
2) Secondly, we design a deep neural network from the solution of the proposed demand-aware bandwidth and power optimization problem. For this, we map the problem's solution into features vectors, from which DL can learn and predict with high accuracy. Then, we combine DL prediction and the formulated optimization problem to obtain lower computational time while allocating the bandwidth and power based on the user demand.

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

3) Finally, we demonstrate the performance of DL-based bandwidth and power allocation via simulation. It shows that the proposed approach has lower computational time while efficiently utilizing the resource of the satellite.

The remainder of the paper is organized as follows. Section II presents the system model. Section III and Section IV discuss the demand-aware bandwidth and power optimization and the DL approach, respectively. The simulation results are presented in Section V. Subsequently, Section VI concludes the paper.

II. System Model

We consider a downlink broadband GEO satellite with N-beams to cover N sub-regions. The satellite is assumed to have a flexible payload that can assign a bandwidth chunk to a specific beam or group of beams from the total available bandwidth B_{total}. Assuming a single user per beam, the total number of possible beam group combinations is given by $M = 2^N - 1$ and the set of groups denoted as $\mathcal{G} = \{G_1, G_2, \ldots, G_m, \ldots, G_M\}$, where G_m is the mth group of beams. The channel vector for the ith beam belonging to the mth group is defined as $\mathbf{h}_m[i] = \left[h_{i,m}[1], h_{i,m}[2], \ldots, h_{i,m}[|G_m|]\right]^T$, where $|G_m|$ is the cardinality of the mth group. Furthermore, the channel coefficient $h_{i,m}[j]$ is given by $h_{i,m}[j] = \frac{\sqrt{\mathbb{G}_R \mathbb{G}_{i,m}[j]}}{4\pi \frac{d_i}{\lambda}} e^{-\mathtt{j}\phi_{i,m}}$, where $\phi_{i,m}$ is the satellite antenna phase component, λ is the wavelength, $\mathbb{G}_R$ is the user antenna gain, $\mathbb{G}_{i,m}[j]$ denotes the received gain from the jth beam by the ith user and d_i is the slant range between the satellite and the ith user.

Since the GEO satellite is assumed to provide coverage to a high-demand area, we assume that precoding is implemented to boost the spectral efficiency and achieve high data rates. Hence, the corresponding precoding weight vector for the ith beam belonging to the mth group is denoted as $\mathbf{w}_m[i] = \left[w_{i,m}[1], w_{i,m}[2], \ldots, w_{i,m}[|G_m|]\right]^T$. We obtain $\mathbf{w}_m[i]$ from the widely-used precoding matrix $\mathbf{W}_m$ based on Minimum Mean Square Error (MMSE) precoding technique i.e. $\mathbf{W}_m = \eta \hat{\mathbf{W}}_m$ with $\hat{\mathbf{W}}_m = \mathbf{H}_m^H (\mathbf{H}_m \mathbf{H}_m^H + \beta \mathbf{I})^{-1}$, and $\eta = \sqrt{\frac{P_m |G_m|}{\text{Trace}\{\hat{\mathbf{W}}_m \hat{\mathbf{W}}_m^H\}}}$. Where β is the regularization factor given by $\beta = N_0 B_m / P_m$, N_0 is the noise density in [W/Hz], B_m is the bandwidth chunk assigned to the mth group and P_m is the maximum transmit power allowed by any beam of the mth group. The signal-to-interference-plus-noise ratio of the ith user in the mth group is given by

$$\gamma_m[i] = \frac{|\mathbf{h}_m^T[i] \mathbf{w}_m[i]|^2}{\sum_{j=1, j\neq i}^{|G_m|} |\mathbf{h}_m^T[i] \mathbf{w}_m[j]|^2 + N_0 B_m}. \tag{1}$$

Hence, the Shannon capacity for the ith user in the mth group is

$$C_m[i] = B_m \log_2(1 + \gamma_m[i]). \tag{2}$$

Subsequently, the overall offered capacity by the system to the ith user is provided as

$$C[i] = \sum_{G_m \in \mathcal{G},\ i \in G_m} C_m[i]. \tag{3}$$

Hence, for user demand $D[i]$, the normalized unmet system capacity is

$$C_{unmet} = \sum_{i=1}^{N} \max(1 - \frac{C[i]}{D[i]}, 0). \tag{4}$$

III. Demand-aware bandwidth and power optimization

In order to maximize the demand satisfaction, the optimal set of groups from the set $\mathcal{G}$ as well as the corresponding bandwidth and power assignment need to be determined. We formulate the following optimization problem with a utility function to minimize: (1) the unmet capacity; and (2) the overall bandwidth consumption.

$$\begin{aligned} \underset{B_m, P_m \forall_m}{\texttt{minimize}} \quad & \sum_{i=1}^{N} \max(1 - \frac{C[i]}{D[i]}, 0) + \sum_{G_m \in \mathcal{G}} \frac{B_m}{B_{\text{total}}} \\ \texttt{s.t.} \quad & T1: \sum_{G_m \in \mathcal{G}} B_m \leq B_{\text{total}}, \\ & T2: B_m \geq 0, \forall_m, \\ & T3: \sum_{G_m \in \mathcal{G}} |G_m| P_m \leq P_{\text{total}}, \\ & T4: P_m \geq 0, \forall_m. \end{aligned} \tag{5}$$

The constraint $T1$ limits the overall of bandwidth allocation of the system from exceeding the total available system bandwidth. Furthermore, $T2$ indicates that B_m should be non-negative. The constraint $T3$ assures that the total transmit power of the system should not exceed the total available system power P_{total}. Additionally, $T4$ constraint indicates that P_m should be non-negative.

The non-linearity of $\gamma_m[i]$ makes (5) non-convex. Hence, convexification is required. To convexify $\gamma_m[i]$, we assume that the power spectral density S_{psd} [W/Hz] is given and thus $\gamma_m[i]$ is expressed in terms of power spectral density. Therefore, with $P_m = S_{\text{psd}} B_m$, (1) is reformulated as

$$\gamma_m[i] = \frac{|\mathbf{h}_m^T[i] \widetilde{\mathbf{w}}_m[i]|^2}{\sum_{j=1, j\neq i}^{|G_m|} |\mathbf{h}_m^T[i] \widetilde{\mathbf{w}}_m[j]|^2 + N_0}, \tag{6}$$

where $\widetilde{\mathbf{w}}_m[i] = \frac{\mathbf{w}_m[i]}{\sqrt{B_m}}$ is the precoding vector in terms of power spectral density. Moreover, $T3$ is re-written as $\sum_{G_m \in \mathcal{G}} |G_m| S_{\text{psd}} B_m \leq P_{\text{total}}$.

Unfortunately, the non-differentiability of the unmet system capacity still hinders from solving the problem (5) directly. Hence, to avoid the non-differentiability, we replace the max function $\max(1 - \frac{C[i]}{D[i]}, 0)$ by upper bound slack variable $u[i]$, $\forall_i$, where $u[i] \geq 0, \forall_i$ and $u[i] \geq 1 - \frac{C[i]}{D[i]}, \forall_i$. Finally, the sub-optimal solution of (5) is reformulated as

$$\begin{aligned} \underset{B_m, \forall_m, u[i], \forall_i}{\texttt{minimize}} \quad & \sum_{i=1}^{N} u[i] + \sum_{G_m \in \mathcal{G}} \frac{B_m}{B_{\text{total}}} \\ \texttt{s.t.} \quad & T1, T2, T3 \end{aligned} \tag{7}$$

$$T5: 1 - \frac{C[i]}{D[i]} \leq u[i], \forall_i$$
$$T6: u[i] \geq 0, \forall_i.$$

Problem (7) is a linear program that can be solved optimally by Simplex Algorithm (SA) [13]. However, the input optimization parameter $B = \{B_1, B_2, \ldots, B_m, \ldots, B_M\}$ of (7) increases exponentially $(M = 2^N - 1)$ as the number of beams increases. Consequently, the complexity of solving (7) increases. In the following section, we propose a DL based acceleration for solving (7).

IV. DL ACCELERATION APPROACH

By analyzing the solution characteristics of (7), it is possible to learn which are the groups with more probability to appear in the solution. Accordingly, we reduce the size of $\mathcal{G}$ and only consider a small number of optimization parameters B_m corresponding to the selected groups. Hence, (7) requires less time to solve as the search space is considerably reduced. In this context, we apply a deep neural network to learn from the optimal solution of (7) how to reduce the group size of $\mathcal{G}$.

A. Single-Model DL (SMDL)

This DL approach uses a single model feedforward neural network as shown in Fig. 1, which learns from the optimal solution of (7) to reduce the size of $\mathcal{G}$. The solution of (7) should be mapped into feature vectors in order to train the network. For this, we first categorize the optimal selected groups of (7) based on their cardinality. For instance, for $N = 4$, if optimal groups are $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, and $\{1,2,3\}$, then, the first three groups are labeled as 2, while the fourth group is labeled as 3. Therefore, we can define the feature vector $X = [x_1, x_2, \ldots, x_n, \ldots, x_N]$ as indicating if groups with the n-cardinality are selected $(x_n = 1)$ or not selected $(x_n = 0)$. In the example above, the feature vector is $X = [0,1,1,0]$, which indicates that an optimal solution is found in the groups of beams with 2-cardinality and 3-cardinality.

By solving (7) for L different demand realizations, we can create L feature vectors. We obtain the demand realizations for $N = 20$, $N = 21$, and $N = 22$ from the demand distribution of Fig. 2. For this distribution, we assume low demand is $5d$ Mbps, moderate demand is $150d$ Mbps, and high demand is $430d$ Mbps, with $d \in [1, 28]$ as the integer value index. With the generated demand realizations $D \in \mathbb{R}^{L \times N}$ and their corresponding feature vectors $X \in \mathbb{R}^{L \times N}$, we can train the deep neural network to predict the feature vector for the new demand realization. Then, based on the predicted vector, we can select groups from $\mathcal{G}$ to solve (7). As an example, for $N = 21$ and demand $D = [280, 20, 50, 290, 35, 130, 400, 360, 420, 530, 410, 35, 30, 120, 810, 770, 420, 160, 150, 140, 70]$ Mbps, the predicted feature vector is $X = [1,1,1,0, 0,0,0, 0,0,0,0,0,0,0, 0,0,0,0,0,0,0]$. Accordingly, 21, 210, and 1330 groups with cardinality of 1, 2, and 3, respectively, are selected. In this case, we perform the optimization using only 1561 groups rather than $\mathcal{G}$ with $2^N - 1 = 2097151$ groups. Consequently, we solve (7) by SA to select the optimal groups in less time.

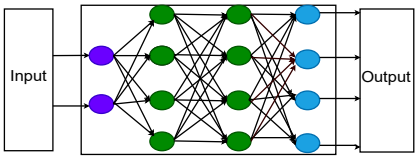


Fig. 1. Single-Model DL

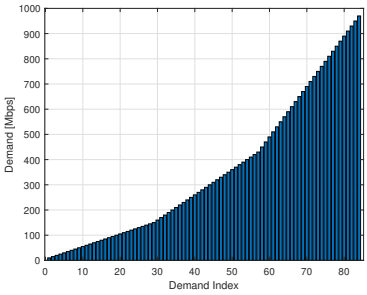


Fig. 2. Demand distribution

B. Multiple-Model DL (MMDL)

The predicted feature vector obtained by a single-model DL approach is used to reduce the size of the $\mathcal{G}$, resulting in a faster solution to (7). However, for some predicted feature vectors, the computational time required to solve (7) may not be significantly different from the time required to solve the original problem. Consider, for example, the single-model DL predicted vector $X = [0,0,0, 1,1,1,1, 1,1,1,1,1,1,1, 0,0,0,0,0,0,0]$. Accordingly, 5985, 20349, 54264, 116280, 203490, 293930, 352716, 352716, 293930, and 203490 with cardinalites of 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14, respectively, are obtained. Hence, a total of 1897150 groups are selected compared to $2^N - 1 = 2097151$ size of $\mathcal{G}$. In this scenario, 90% of the groups in $\mathcal{G}$ are selected, and the computational time required by (7) is not significantly less than the original problem with the input size $\mathcal{G}$. This problem can be addressed by reducing the group size of each cardinality. Hence, we propose a multiple-model DL approach, depicted in Fig. 3, in which a primary neural network (single-model DL) is used to predict the optimal cardinality of the groups, whereas secondary neural networks are used to reduce the number of groups for each cardinality. However, secondary neural networks may not be required in some cases where the cardinality group size is small. Hence, we model the secondary neural network only for the group size of n-cardinality that satisfies the minimum threshold group size value of Γ. For this network, we prepare the future vector as follows:

- Firstly, we generate all possible groups for the n-cardinality. For example, for $N = 4$, and $n = 2$, the possible groups are $\{1,2\}$,$\{2,3\}$,$\{2,4\}$,$\{1,3\}$,$\{3,4\}$,$\{1,4\}$. Note that, all elements within a group need to be arranged in increasing order.
- Secondly, we put the groups in sub-sets according to their first element. For instance, the groups $\{1,2\}$, $\{1,3\}$,

and $\{1,4\}$ are grouped together since the first element is 1. Accordingly, the group category is given by $l_1 = \{\{1,2\},\{1,3\},\{1,4\}\}$, $l_2 = \{\{2,3\},\{2,4\}\}$ and $l_3 = \{\{3,4\}\}$.

- Finally, we encode the category groups into a binary valued vector $Z = [z_1, z_2, \ldots, z_j, \ldots, z_{N-n+1}]$, where $z_j = 1$ indicates any group from l_j is a possible solution to (7). For example, if (7) selects the optimal groups $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, then the feature vector for the 2-cardinality is $Z = [1,0,0]$.

Then, the neural network for the n-cardinality is trained using the feature vector Z and the corresponding demand D obtained using Fig. 2. Note that the second neural network is only activated if the primary neural network output for x_n is $x_n = 1$ and the n-cardinality group size is above the threshold value Γ. From this property, the Multiple-Model DL approach predicts each element of the feature vector from the primary neural network if its output $x_n = 0$ or if the n-cardinality group size is below the threshold value Γ, otherwise, it is obtained from the second neural network.

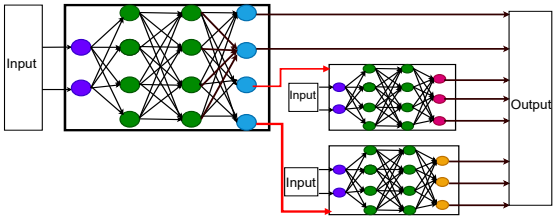


Fig. 3. Multiple-Model DL

V. Simulation Results

We consider $N = 20$, $N = 21$, and $N = 22$ beams to evaluate the proposed DL approach. The beams are generated by a Direct Radiating Antenna (DRA), with elements spaced by 5λ apart and provided by the European Space Agency (ESA). Additionally, we assume a single user is located at the center of each beams. Furthermore, we consider a fully-connected neural network[1] for the proposed methods, and the parameters required to model the network are summarized in Table I. Additionally, the data sets required to train (80% of the data) and test (20% of the data) the network are provided in Table II. We compare the proposed methods SA+SMDL and SA+MMDL with the optimal solver of SA.

Table III shows the average computational time of the SA, the SA+SMDL, and the SA+MMDL. We observe that the proposed SA+SMDL and SA+MMDL methods require less computational time than the SA algorithm. For instance, for $N = 20$, the computational time of SA+SMDL and SA+MMDL is 24.7525 s and 10.8427 s, respectively, whereas the time for SA is 34.1118 s. Similarly, for $N = 21$, the SA+SMDL and the SA+MMDL need 44.2976 s and 21.3881 s, respectively, to solve (7). In contrast, SA requires 87.0577 s to solve (7). Furthermore, for $N = 22$, both the SA+SMDL and the SA+MMDL reduce the computation of the original problem by 25.67% and 65.96%, respectively. Therefore, the proposed methods require less computational time than the original problem.

[1]Note that the predicted values are rounded to zero if they are less than 0.5, otherwise to one.

TABLE I
System Parameters

Satellite Orbit	13°E
Satellite Beam Pattern	Provided by ESA
Number of beams (N)	$20, 21, 22$
System Bandwidth (B_{tot})	500 MHz
Noise power density (N_0)	-204 dBW/Hz
Max. beam gain ($G_i[j]$)	51.8 dBi
User antenna gain (G_R)	39.8 dBi
Power spectral density (S_{spd})	-78.8 dBW/Hz
Total available transmit power (P_{total})	132 W
Number of hidden layers	4
Nodes per hidden layer	200
Number of output nodes	Primary (N) and Secondary ($N-n-1$)
Number of input nodes	Primary (N) and Secondary (N)
Activation function	Relu
Optimizer	Adam [14]
Loss function	Root Mean Squared Error (RMSE)
Batch size	1024
Batch normalization	per layer
Number of epochs	128
Minimum cardinality size (Γ)	15000
Mathematical optimization solver to (7)	Simplex Algorithm (SA) using Mosek solver
Single-Model DL	MATLAB Deep Learning Toolbox
Multiple-Model DL	MATLAB Deep Learning Toolbox

TABLE II
The Size of the Data Set

DL	$N = 20$	$N = 21$	$N = 22$
Primary	60000	60000	60000
Secondary $n = 5$	26045	22275	19560
Secondary $n = 6$	39270	35855	32475
Secondary $n = 7$	46035	44440	42470
Secondary $n = 8$	45820	46335	46195
Secondary $n = 9$	39400	41615	43790
Secondary $n = 10$	34240	36070	38410
Secondary $n = 11$	35530	37365	37000
Secondary $n = 12$	34500	37060	38340
Secondary $n = 13$	22300	27075	33140
Secondary $n = 14$	9330	14465	20785
Secondary $n = 15$	-	5810	9520
Secondary $n = 16$	-	1515	3265
Secondary $n = 17$	-	-	985

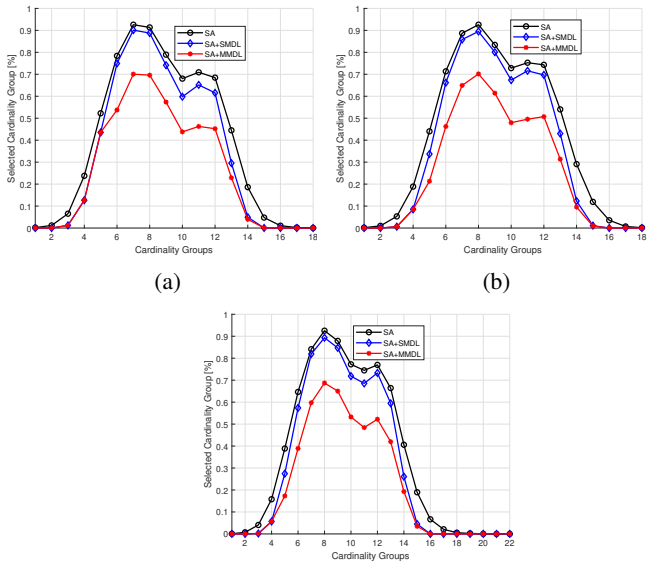
TABLE III
Computational Time

Beams	SA	SA+SMDL	SA+MMDL
20	34.1118 s	24.7525 s	10.8427 s
21	87.0577 s	44.2976 s	21.3881 s
22	165.949 s	123.3419 s	56.4812 s

Fig. 4 shows the cardinality prediction performance of SA+SMDL and SA+MMDL compared with the optimal cardinality[2] of the SA for $N = 20$, $N = 21$ and $N = 22$. We observe that SA+SMDL and SA+MMDL have the same prediction for cardinality below 5 and above 17. This same prediction results from both methods use the same primary neural network to predicate cardinality groups below 5 and above 17. For the cardinality between 5 and 15, the SA+SMDL shows a better prediction than the SA+MMDL. For example, in Fig. 3a for $N = 20$ with cardinality 8, the SA+SMDL and SA+MMDL predicates 97% and 78% of the SA optimal cardinality, respectively. In addition, Fig. 3b for $N = 21$

[2]Note that, in Fig. 4, the number of Selected Cardinality Groups (SCG) is determined by dividing the total number of n-cardinality selected from the test data by the total size of the test data. Furthermore, the predication performance of SA+SMDL and SA+MMDL in percent is given by $\frac{\text{SCG of SA+SMDL[\%]}}{\text{SCG of SA[\%]}} \times 100$ and $\frac{\text{SCG of SA+MMDL[\%]}}{\text{SCG of SA[\%]}} \times 100$, respectively.

with cardinality 9, the SA+SMDL and SA+MMDL predicates 96% and 77% of the optimal cardinality of the SA, respectively. Similarly, Fig. 3c for $N = 22$, the SA+SMDL predicates 93% of SA optimal solution of cardinality 10, while SA+MMDL predicates 74% of it. The SA+MMDL prediction is less accurate than SA+SMDL due to the second neural network, which can still give an incorrect prediction even though the primary is correct. Generally, SA+SMDL provides more accurate predictions than SA+MMDL at the expense of longer computation times.



(a) (b)

(c)

Fig. 4. Comparison cardinality group predication of SA+SMDL and SA+MMDL with optimal cardinality group selection of SA: (a) $N = 20$; (b) $N = 21$; (c) $N = 22$.

Table IV shows the average bandwidth usage[3] of SA, SA+SMDL and SA+MMDL. For $N = 20$, the bandwidth usage of SA+SMDL and SA+MMDL is higher than SA by 77.2 KHz and 189.7 kHz, respectively. Furthermore, for $N = 21$, the SA+SMDL and SA+MMDL need 69.6 KHz and 179.3 kHz in addition to SA bandwidth usage. Similarly, for $N = 22$, the bandwidth usage of SA+SMDL exceeds SA by 135kHz. In contrast, the bandwidth usage of SA+MMDL exceeds SA by 275.8 kHz. We observe that the proposed methods consume more bandwidth than SA due to cardinality prediction errors. In this case, the optimization requires more bandwidth to satisfy the demand per beam. However, the proposed methods require substantially less computation time than SA.

VI. Conclusions

We propose a demand-aware bandwidth and power allocation algorithm for GEO satellites based on a combination of analytical optimization and DL. Thus, analytical optimization enables bandwidth and power allocation, while DL can speed up computation. Future work will focus on developing an efficient feature vector for the DL approach. This will reduce the original problem computation time even further while adequately utilizing the system's bandwidth.

[3]Note that power allocation is calculated based on bandwidth using the formula $P_m = S_{\text{psd}} B_m$. Hence, we can draw the same conclusion regarding power consumption as we did about bandwidth usage.

TABLE IV
Bandwidth Usage

Beams	SA	SA+SMDL	SA+MMDL
20	361.2951 MHz	361.3723 MHz	361.4848 MHz
21	363.8607 MHz	363.9203 MHz	364.04 MHz
22	365.6942 MHz	365.8292 MHz	365.97 MHz

Acknowledgment

Some of the results have been obtained using the Luxembourgish MeluXina supercomputer, in the context of the first call for early access. For this, we thank MeluXina providers.

References

[1] S. Kisseleff, E. Lagunas, T. S. Abdu, S. Chatzinotas, and B. Ottersten, "Radio resource management techniques for multibeam satellite systems," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2448–2452, Aug 2021.

[2] C. N. Efrem and A. D. Panagopoulos, "Dynamic Energy-Efficient Power Allocation in Multibeam Satellite Systems," *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 228–231, 2020.

[3] T. S. Abdu, S. Kisseleff, E. Lagunas, and S. Chatzinotas, "Flexible Resource Optimization for GEO Multibeam Satellite Communication System," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[4] A. I. Aravanis, B. Shankar M. R., P. Arapoglou, G. Danoy, P. G. Cottis, and B. Ottersten, "Power Allocation in Multibeam Satellite Systems: A Two-Stage Multi-Objective Optimization," *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3171–3182, June 2015.

[5] 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 Transactions on Broadcasting*, vol. 64, no. 2, pp. 266–280, Jun. 2018.

[6] J. J. G. Luis, N. Pachler, M. Guerster, I. del Portillo, E. Crawley, and B. Cameron, "Artificial intelligence algorithms for power allocation in high throughput satellites: A comparison," in *2020 IEEE Aerospace Conference*, 2020, pp. 1–15.

[7] 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 Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2398–2408, 2018.

[8] X. Hu, X. Liao, Z. Liu, S. Liu, X. Ding, M. Helaoui, W. Wang, and F. M. Ghannouchi, "Multi-agent deep reinforcement learning-based flexible satellite payload for mobile terminals," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9849–9865, 2020.

[9] 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 Systems Journal*, pp. 1–12, 2020.

[10] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. C. Reinhart, and D. J. Mortensen, "Multiobjective reinforcement learning for cognitive satellite communications using deep neural network ensembles," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 5, pp. 1030–1041, 2018.

[11] L. Lei, L. You, Q. He, T. X. Vu, S. Chatzinotas, D. Yuan, and B. Ottersten, "Learning-Assisted Optimization for Energy-Efficient Scheduling in Deadline-Aware NOMA Systems," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 615–627, 2019.

[12] M. Vázquez, P. Henarejos, I. Pappalardo, E. Grechi, J. Fort, J. C. Gil, and R. M. Lancellotti, "Machine learning for satellite communications operations," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 22–27, 2021.

[13] K. Murty, *Linear programming*. NJ, USA: Wiley, 1983.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.