Deep Learning for Beam Hopping in Multibeam Satellite Systems

Lei Lei, Eva Lagunas, Yaxiong Yuan, Mirza Golam Kibria, Symeon Chatzinotas, and Björn Ottersten

Interdisciplinary Centre for Security, Reliability and Trust, Luxembourg University, Luxembourg

{lei.lei; eva.lagunas; yaxiong.yuan; mirza.kibria; symeon.chatzinotas; bjorn.ottersten}@uni.lu

***Abstract*—Data-driven approaches, e.g., deep learning (DL), have been widely studied in terrestrial wireless communications fields, proving the benefits and potentials of such techniques. In comparison, DL for satellite networks is studied to a limited extent in the literature. In this paper, we develop a DL assisted approach to facilitate efficient beam hopping (BH) in multibeam satellite systems. BH is adopted to provide a high level of flexibility to manage irregular and time variant traffic requests in the satellite coverage area. Conventional iterative optimization approaches and typical data-driven techniques may have their respective limitations in achieving timely and satisfactory performance. We herein explore a combined learning-and-optimization approach to provide a fast, feasible, and near-optimal solution for BH scheduling. Numerical study shows that in the proposed solution, the learning component is able to largely accelerate the procedure of BH pattern selection and allocation, while the optimization component can guarantee the solution's feasibility and improve the overall performance.**

***Index Terms*—Beam hopping, deep learning, optimization, satellite communications.**

I. Introduction

Satellite resources are expensive and thus it is necessary to optimize and, whenever possible, share these precious resources. The main objective of the present work is to assess the use of data-driven approaches, e.g., deep learning (DL), to optimize the satellite resource allocation in multibeam system scenarios with non-uniform traffic demands. The capability to flexibly allocate on-board resources over the service coverage is becoming a must for future broadband multibeam satellites [1]. Previous and current systems have shown that in large multibeam satellites the demand in some spot beams greatly exceeds the available capacity (hot spots) while in others the situation is inverted (cold spots) [1], [2]. This raises a paradoxical scenario where demand is left unmet in the hot spots while capacity is left unused in the cold spots. The consequence for the satellite operator and service provider is twofold: a loss of the revenue corresponding to the unmet demand, and the loss of the investment in the unused capacity. The primary goal of flexibility is then to solve this paradox and maximize the amount of system capacity that is actually used (sold) by allocating it where needed. While in conventional payloads the resources per beam are fixed and uniformly distributed across beams [3], therefore providing the same capacity to each beam.

Satellite's payload flexibility can be implemented in several ways involving the flexible allocation of bandwidth (irregular frequency reuse), of time, and of power. Beam hopping (BH) for satellite systems has been proposed as a promising technological enabler to provide a high level of flexibility to manage irregular and time variant traffic requests in the satellite coverage area [1], [4]. With BH, all the available satellite resources are employed to provide service to a certain subset of beams, which is active for some portion of time, dwelling just long enough to fill the demand in each beam. The set of illuminated beams changes in each time-slot based on a time-space transmission pattern (called *BH illumination pattern*) that is periodically repeated.

The challenging task in BH systems is to *determine the beam illumination pattern*, i.e., the beams to be simultaneously activated and for how long. Some works in the literature have addressed the beam illumination pattern design from different perspectives, e.g., iterative algorithm for BH illumination design in [2], BH for power minimization in [5]. The main difficulty in BH pattern design is the exponentially increased search space for obtaining optimal BH patterns. For the satellite systems composed of hundreds/thousands of beams, it renders a complicated optimization procedure with long computation time. Given the inherent difficulty in optimal BH design, it is hard to expect that a suboptimal solution can achieve satisfactory performance and with very low complexity [6]. In this context, DL appears as a promising technique that offers an alternative to design efficient algorithms for complex resource management in wireless networks. DL has received considerable research attention over the past few years in wireless communications [6]–[8].

In the literature, applying DL to satellite scenarios is studied to a limited extent compared to its applications in terrestrial communication systems. In addition, suitable DL based solutions for BH pattern design, along with their performance evaluation have not been investigated yet for satellite scenarios. Being aware of this gap, we provide initiative investigations for DL in BH optimization. The main contributions of this work are summarized as follows.

- We explore a viable way to combine learning and optimization methods for BH design, in order to overcome the shortcomings in conventional optimal/suboptimal algorithms and classical learning models.
- To apply DL to BH optimization, we identify a learnable feature for BH, i.e., the cardinality of the beam patterns, which leads to high prediction accuracy in DL. In the

proposed algorithm, we provide a simple but efficient manner to combine DL and optimization components.

- We carry out performance comparisons among optimal, suboptimal, and the proposed DL-based optimization (DBO) algorithm, in terms of computational time and optimality approximation. Numerical study demonstrates that the learning component is capable of dramatically accelerating the procedure of promising BH pattern selection. While the optimization component can guarantee the solution's feasibility and improve the overall performance.

II. System Model

A. Multibeam Satellite Systems

We consider the forward link of a broadband multibeam satellite system that aggressively reuses frequency resources. We consider a bent-pipe transparent geostationary orbit (GEO) satellite architecture, which relays the signal from the gateway to the final receivers. We assume the forward link transmission of N satellite beams, which are considered to be equal to the number of transmitting elements on the satellite. All beams share the same frequency band B. The key notations are summarized in Table I.

Table I
Notations

N	number of beams
n	beam index $n = 1, \dots, N$
B	bandwidth per beam
D_n	requested demand in beam n
T_H	duration of a BH cycle
g	index of snapshots, $g = 1, \dots, G$
$\mathcal{N}_g$	set of all the active beams in snapshot g
$\mathcal{G}$	set of candidate snapshots, $\lvert\mathcal{G}\rvert = G$
t_g	duration of snapshot g (continuous or discrete)
R_n	offered capacity for beam n
R_{ng}	achievable rate of beam n in snapshot g
$h_{j,n}$	channel gain from the j-th satellite antenna to beam n
p_n	transmit power for beam n

The channel matrix $\mathbf{H}$ gathers the forward link budget information and phase rotations introduced by the over-the-air propagation. In particular,

$$\mathbf{H} = \mathbf{P}\hat{\mathbf{H}} \tag{1}$$

where the matrix $\mathbf{P}$ models the phase variations due to the different propagation paths and its components $[\mathbf{P}]_{x,y}$ are defined as,

$$[\mathbf{P}]_{x,y} = \begin{cases} e^{j\phi_x} & \text{if } x = y \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

being ϕ_x a uniform random variable between $-\pi$ and π.

The matrix $\hat{\mathbf{H}}$ represents the real channel state information (CSI) contribution, which is determined by the satellite antenna gain, the path loss, the received antenna gain and the noise power. More precisely, the (k,n)-th component of $\hat{\mathbf{H}}$ is given by,

$$\left[\hat{\mathbf{H}}\right]_{k,n} = \frac{\sqrt{G_R G_{k,n}}}{4\pi \frac{d_k}{\lambda} \sqrt{K_B T B}} \tag{3}$$

where G_R is the user terminal antenna gain, $G_{k,n}$ denotes the gain from the n-th satellite antenna towards the k-th user served within the n-th beam and d_k is the slant range between the satellite and the k-th user. The term $\sqrt{K_B T B}$ represents the noise contribution, where K_B is the Boltzmann constant and T is the receiver noise temperature. It is common practice to include the noise contribution into the channel model in order to proceed with the assumption of unit-variance noise.

The received signal at the k-th user located at the n-th beam can be expressed as,

$$y_{k,n} = \mathbf{h}_{k,n}^T \mathbf{x} + n_{k,n}, \tag{4}$$

where $\mathbf{h}_{k,n} \in \mathbb{C}^{N\times 1}$ is the CSI vector corresponding to this particular user, $\mathbf{x} \in \mathbb{C}^{N\times 1}$ represents the vector of N symbols and $n_{k,n}$ is the complex Additive White Gaussian Noise (AWGN) at user k of beam n.

For the sake of clarity, we can rearrange the received signals (4) by using the following matrix notation,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{5}$$

The received samples are arranged into $\mathbf{y} = \left[\mathbf{y}_1^T \quad \mathbf{y}_2^T \quad \cdots \quad \mathbf{y}_N^T\right]^T$, where $\mathbf{y}_n$ is the vector containing the received signal for the users belonging to the n-th beam.

B. Beam Hopping

The main design task in BH is to determine which beams to be illuminated together and the duration. The BH illumination pattern design also consists of designing a BH cycle of duration T_H, which is periodically repeated. Within a BH cycle, a number of illuminated *snapshots* are used. In this paper, we define a snapshot as a particular arrangement of illuminated and un-illuminated beams. As an example, Fig. 1 illustrates a BH cycle composed of 6 different snapshots. Each beam is covered by the used snapshots at least once. These selected snapshots can be sequentially scheduled with optimized duration.

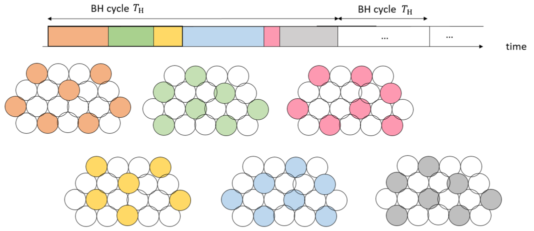

Figure 1. An example of a BH cycle composed of 6 different snapshots

By enumeration, the total number of possible snapshots, denoted as G, is equal to 2^N. Clearly, the number of snapshots increases exponentially with the number of beams, resulting in a huge search space for optimization. We define that a BH cycle is segmented into N_{slot} time slots of duration T_{slot} (i.e., $T_H = T_{slot} N_{slot}$). The division of the BH cycle into time-slots is depicted in Fig. 2, where the selected snapshots and its duration are illustrated using different colors. We use variables $t_1, \dots, t_g, \dots, t_G$ to represents the number of slots allocated to snapshots $1, \ldots, g, \ldots, G$, respectively. If t_g is equal to 0, then the g-th snapshot is not used. Since each time slot can only be used by at most one snapshot, thus $\sum_{g \in \mathcal{G}} t_g T_{slot} = T_H$.

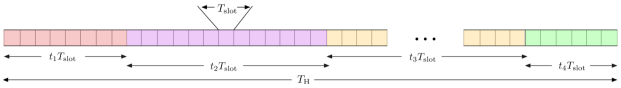

Figure 2. A BH Frame

A beam provides service to multiple terminal users in the coverage area. In a beam, multiple terminal users are assumed to be served by the time division multiple access (TDMA) approach. In this paper, we focus on the long-term BH performance at the beam level, and thus, we use beam's demand D_n to represent the aggregated users' demands in the n-th beam. Correspondingly, we assume a single user virtually located in the center of the beam that emulates the overall traffic demand of the corresponding beam. Then, the channel matrix $\mathbf{H}$ can be simplified to a $N \times N$ matrix, where the diagonal elements $h_{n,n}$ represent the channel gain from the n-th satellite antenna to the user in beam n. The delivered data rate of beam n in snapshot g in a time slot can be expressed as, $R_{ng} = T_{slot} B f_{DVB} \left(\frac{p_n h_{n,n}}{\sum_{j \in \mathcal{N}_g \setminus \{n\}} p_j h_{jn} + \sigma^2} \right)$, where f_{DVB} is the rate mapping function according to the digital video broadcasting through satellite second generation specifications extensions (DVB-S2X) [3].

III. Problem Formulation

The BH design problem is formulated in P0. The problem aims at optimizing the performance of offered capacity to requested demand ratio (OCDR), i.e., the fraction of the offered capacity (R_n) divided by the requested demand (D_n) of the worst beam. As a consequence of BH optimization, the offered and requested capacity can achieve a good match among beams. The optimization task is to determine which snapshots to be scheduled in a BH cycle, and how many time slots to be used for each snapshot. The constraint (6b) states that the total duration for the illuminated snapshots should be equal to a BH cycle T_H. Constraints (6c) define the offered capacity for each beam. In (6d), the optimization variables $t_1, \ldots, t_G$ are integer, resulting in a mixed integer linear programming problem (MILP) in P0.

$$\text{P0:} \quad \max_{t_1,\ldots,t_G} \min \left(\frac{R_1}{D_1}, \ldots, \frac{R_N}{D_N}\right) \tag{6a}$$

$$\text{s.t.} \sum_{g \in \mathcal{G}} t_g T_{slot} = T_H, \tag{6b}$$

$$R_n = \sum_{g \in \mathcal{G}} t_g T_{slot} B f_{DVB} (1 + \frac{p_n h_{n,n}}{\sum_{j \in \mathcal{N}_g \setminus \{n\}} p_j h_{jn} + \sigma^2}), \forall n \tag{6c}$$

$$t_1, \ldots, t_G, \text{ i}nteger \tag{6d}$$

Solving P0 is difficult in general, in particular for the large-sale instances. The high computational complexity and long computing time impose obstacles for real-time BH scheduling. We circumvent this issue by solving P0's linear relaxation problem which is formulated in P1 with non-negative continuous variables $t_1, \ldots, t_G$ representing the duration of snapshots $1, \ldots, G$, respectively.

$$\text{P1:} \quad \max_{t_1,\ldots,t_G} \min \left(\frac{R_1}{D_1}, \ldots, \frac{R_N}{D_N}\right) \tag{7a}$$

$$\text{s.t.} \sum_{g \in \mathcal{G}} t_g = T_H, \tag{7b}$$

$$R_n = \sum_{g \in \mathcal{G}} t_g B f_{DVB} (1 + \frac{p_n h_{n,n}}{\sum_{j \in \mathcal{N}_g \setminus \{n\}} p_j h_{jn} + \sigma^2}), \ \forall n \tag{7c}$$

$$t_1, \ldots, t_G \geq 0. \tag{7d}$$

P1 can be equivalently reformulated as P1' by introducing an auxiliary variable η. From P1', one can observe that P1 is a Linear Programming (LP) problem.

$$\text{P1':} \quad \max_{t_1,\ldots,t_G} \eta \tag{8a}$$

$$\text{s.t.}(7b), \ (7c), \ (7d) \tag{8b}$$

$$\frac{R_n}{D_n} \geq \eta, \forall n = 1, \ldots, N, \tag{8c}$$

In general, P1 (or P1') provides an upper bound for P0. However, the optimal solution of the LP P1 can ultimately approach to the MILP P0, when the granularity of time slots improves, e.g., reduce the duration T_{slot}. Once T_{slot} is small enough, e.g., $T_{slot} = 1 \ ms$ ($0.001 \ s$), P0 and P1 converge to almost the same optimum points (refer to Fig. 4 in Section V). As a result, the training data in DL can be generated more efficiently by solving P1 rather than P0.

IV. The Proposed DBO Algorithm

To optimally solve P0 and P1, some standard optimization methods can be applied, e.g., branch-and-bound (B&B) algorithm for MILP P0, and simplex algorithm (SA) or column generation (CG) algorithm for LP P1 [9]. The difficulty is that the size of candidate snapshot set $\mathcal{G}$ or number of variables in P0 and P1 increase exponentially with the number of beams, e.g., 2^N snapshots in $\mathcal{G}$. When larger-scale scenarios are considered, e.g., more beams, the number of variables in optimization problems P0 and P1 becomes huge. The computational complexity and time are not affordable for practical BH scheduling. From the learning aspect, classical end-to-end learning approaches, i.e., relying on a learning model to directly output a complete solution for the addressed optimization problem, may help but typically be applicable to limited instances [6], [10], [11] .

By observing and analyzing the optimal solutions from P1, we extract and characterize the following feature which can be well learned by DL. The feature vector consists of N binary elements, $\boldsymbol{v} = [v_1, \ldots, v_n, \ldots, v_N]$, where v_n represents if any of n-cardinality snapshots (n active beams in the snapshot) is scheduled in the optimum ($v_n = 1$), or none of n-cardinality snapshots is used at all ($v_n = 0$). For example, in 4-beam BH, if the optimal snapshots are $\{\{1, 2\}, \{1, 3, 4\}, \{2, 3\}\}$, the corresponding vector $\boldsymbol{v}$ is organized as $[0, 1, 1, 0]$ since only 2-cardinality and 3-cardinality snapshots are scheduled.

In DL training, a training/validation/test set contains two parts data, i.e., input data and optimized results. The input data includes channel matrix $\boldsymbol{H}$, traffic demand per beam $D_1, \ldots, D_N$, and the power per beam $p_1, \ldots, p_N$. Note that DL only accepts the real-value inputs, thus the original complex values in $\boldsymbol{H}$ are normalized to real values. The beam coverage area, channel matrix, and traffic demand per beam are generated by an adopted emulator based on real-life topologies [12]. The conventional optimal algorithms, SA or CG, are used to produce the corresponding optimal labels offline. The optimal labels are organized in the feature vector $[v_1, \ldots, v_N]$ for each realization. A DL model is then trained to learn the mapping from the input data to the optimal labels.

After training, the DL model is expected to provide its fast predictions. We show the procedure in Fig. 3. Scanning all the 0-elements in the predicted feature vector, a considerably large amount of snapshots can be excluded from the original candidate set, forming a small-scale candidates set. For instance, if $\boldsymbol{v} = [0, 1, 1, 0]$, all the 1-cardinality and 4-cardinality snapshots will be removed from the restricted set. Note that the raw values in the DL model's output nodes may not be binary. A rounding operation is adopted to convert fractional values to binary. Specifically, we use M as the mean of $\boldsymbol{v}$. If any fractional value $v_n > \alpha M$, we set $v_n = 1$, otherwise zero, where $\alpha > 0$ is a control parameter.

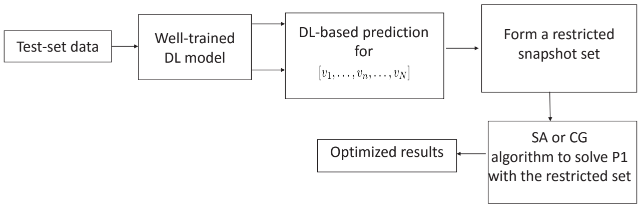
Test-set data
Well-trained DL model
DL-based prediction for $[v_1, \ldots, v_n, \ldots, v_N]$
Form a restricted snapshot set
SA or CG algorithm to solve P1 with the restricted set
Optimized results

Figure 3. DL-based optimization approach in operation phase

V. Numerical Results

To evaluate the performance of the proposed DBO, we consider a multi-beam GEO satellite scenario, whose antenna pattern and link budget parameters have been provided by "ESA71" simulator [12]. We adopt a fully-connected deep neural network (FC-DNN) as the DL model. The adopted parameters in simulation are summarized in Table II. Firstly, we show the optimality gaps between P1 and P0 with respect to T_{slot} in Fig. 4. As expected, under the adopted time scale $T_{slot} = 1\ ms$, the optimality gap between P0 and P1 is diminished. The generated optimal labels in training sets can be used exchangeably by optimally solving P0 or P1. Secondly, we show the performance gain of DBO over optimization algorithms and end-to-end learning. In Table III, the computational time of DBO (consumed time for executing the procedure in Fig. 3) and the optimal methods are compared. The computational time in B&B, SA, and CG exponentially increases with the number of beams, whereas the proposed algorithm is insensitive to the problem's scale. It keeps at the same magnitude in all the cases. In general, the computational time of the proposed DBO is dramatically reduced compared with the iterative optimization algorithms.

Table II
Parameters Settings

Satellite longitude	13^0 (GEO)
Number of beams	16
Carrier frequency	19.5 GHz
Bandwidth per beam	500 MHz
Transmit power per beam	20 dBW
Noise power	-120.64 dBW
BH cycle T_H	256 ms
T_{slot} in P0	1 ms
Number of hidden layers	3
Nodes per hidden layer	200
Number of output nodes	$N = 16$
Activation function	Relu
Optimizer	Adam [10]
Loss function	Mean Squared Error (MSE)
Size of training sets	18000
Size of test sets	2000
Batch size	128
Number of epochs	450
Algorithm 1 implementation	Python + TensorFlow
B&B, SA, and CG algorithms	Python + Gurobi

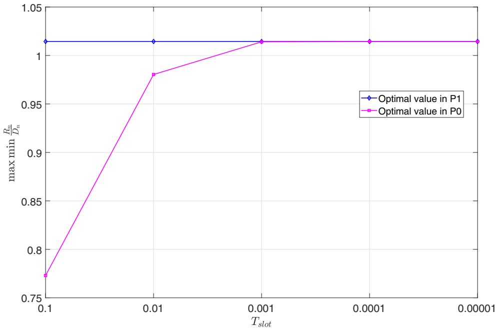

Figure 4. Optimality gaps between P1 and P0 with respect to T_{slot}

Table III
Comparison in Computational Time

Beams	B&B for P0	SA for P1	CG for P1	DBO
$N = 10$	15.094	0.91	0.56	0.025
$N = 16$	563	3.142	2.028	0.076
$N = 20$	>3600	181.4	125.3	0.084
$N = 21$	>3600	412.5	189.4	0.135
$N = 22$	>3600	2318	1303	0.203
$N = 25$	>3600	>3600	>3600	0.391

Fig. 5 shows the prediction accuracy of the FC-DNN in DBO. The metric "prediction accuracy" is derived by comparing the predicted vector $\boldsymbol{v}$ from FC-DNN with the optimal vector $\boldsymbol{v}^*$ from optimal algorithms. The y-axis value stands for the ratio of how many elements in $\boldsymbol{v}$ consistent with the optimum. The prediction accuracy is able to be improved when more training sets are adopted. When training is sufficient, the OCDR values between DBO and the optimum are close, with the average gap around 5%. In contrast, we use an illustrative example in Fig. 6 to show the performance in classical end-to-

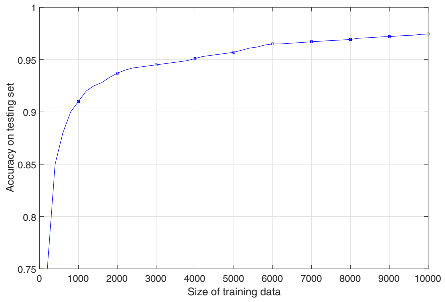

Figure 5. Predict accuracy with respect to training set size

end learning approach. The same FC-DNN is adopted, but the feature vector becomes $[t_1, \ldots, t_G]$. The OCDR values in the optimal solution and the end-to-end learning (ETEL) method are compared in 500 test sets. From the results, the solutions derived by end-to-end leaning are infeasible in almost all the instances due to the imperfect prediction in DNN. In addition, the optimality gaps are considerably large, merely achieving around 50% of the optimal value in average. The choice of which features to be learned is critical for the prediction performance. The results also confirm the necessity of designing learnable features. Lastly, in Fig. 7, we show that

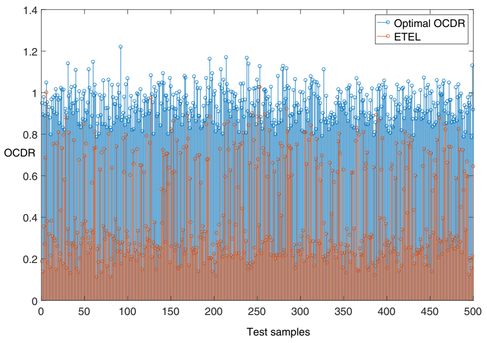

Figure 6. Performance comparison between optimum and end-to-end learning

the accuracy performance can be scaled by parameter α. The accuracy performance can be improved by adopting small α to round the fractional values in DNN's output to binary. DBO is able to approach to the optimum when an appropriate α adopted.

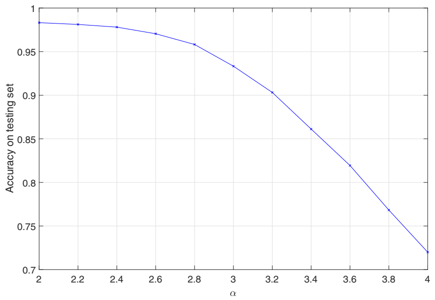

Figure 7. Predict accuracy with respect to parameter α

VI. Conclusion

In this paper, we have applied DL technique to solve an optimization problem in BH illumination pattern design. A DL-based optimization algorithm is proposed to reap the benefits of learning and optimization. We have showed that it is of importance to design a proper learning feature in order to come up with a good prediction in DL. Numerical results have showed that the proposed algorithm can be used to limit the search space of optimization problems, and therefore speed-up the process of obtaining near-optimal solution. In this way, the efficient computations of DL is combined with the high-quality and feasible solution of the optimization approach.

Acknowledgment

This work was supported in part by the European Research Council (ERC) project AGNOSTIC (under Grant 742648), the FNR CORE project ROSETTA (under Grant 11632107), the FNR CORE project FlexSAT, and the ESA project SATNEX IV (Opinions, interpretations, recommendations and conclusions presented in this paper are those of the authors and are not necessarily endorsed by the European Space Agency).

References

[1] A. Freedman, D. Rainish, and Y. Gat, "Beam Hopping: How To Make it Possible," *Ka and Broadband Communication Conference, Bologna, Italy*, Oct. 2015.

[2] J. Lei and M. Vazquez-Castro, "Multibeam Satellite Frequency/Time Duality Study and Capacity Optimization," in proc. IEEE ICC, 2011.

[3] 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.

[4] J. Anzalchi, et al., "Beam Hopping in Multi-Beam Broadband Satellite Systems: System Simulation and Performance Comparison with Non-Hopped Systems," In proc. *Advanced Satellite Multimedia Systems Conference (ASMS)*, pp. 248–255, Sept. 2010.

[5] R. Alegre-Godoy, N. Alagha, and M. Vazquez-Castro, "Offered Capacity Optimization Mechanisms for Multibeam Satellite Systems," in proc. IEEE ICC, Jun. 2012.

[6] F. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, "Learning Radio Resource Management in RANs: Framework, Opportunities, and Challenges," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 138–145, Sept. 2018.

[7] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 28–35, Jun. 2018.

[8] L. Lei, L. You, G. Dai, T. Vu, D. Yuan, and S. Chatzinotas, "A Deep Learning Approach for Optimizing Content Delivering in Cache-Enabled HetNet," in proc. IEEE ISWCS, Aug. 2017.

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

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. London, UK: MIT Press, 2016.

[11] L. Lei et al., "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, Sept. 2019.

[12] "ESA Project FlexPreDem - Demostrator of Precoding Techniques for Flexible Broadband Systems," Online: https://wwwfr.uni.lu/snt/research/sigcom/projects.