Dynamic resource management algorithm reconfiguration for multibeam satellite constellations

Sergi Aliaga, Juan Jose Garau-Luis, Edward Crawley, Bruce Cameron
Massachusetts Institute of Technology
77 Massachusetts Av. 33-409
Cambridge, MA 02139
{aliaga, garau, crawley, bcameron}@mit.edu

Abstract—**Satellite mega constellations are a reality. The new generation of High Throughput Satellites has motivated the research in Dynamic Resource Management (DRM) strategies for satellite communications. Unprecedented levels of flexibility, granted by adjustable payloads able to reallocate resources such as power or frequency in real time, have placed manual resource allocation in a disadvantageous position. Many algorithmic solutions have been specifically proposed to address this issue.**

However, the majority of the proposed models have mostly been proven under conditions that do not represent the upcoming satellite communications scenarios. Failure to scale up those algorithmic solutions to current high-dimensional constellations might result in a poor resource allocation, or even worse, a service agreement violation. In addition, since many of the elements that are input to these algorithms change over time, new requirements impose being able to not only scale up but also reconfigure in real time in order to make the best use of capacity.

To that end, this work presents and tests a methodology to dynamically configure DRM algorithms with the aim of granting viability of operation under multiple possible scenarios that reflect realistic operations. Using the specific frequency assignment problem as a test case, we show that adapting the algorithm's configuration based on analyzing the input scenario results in up to 79% reduction in computing time, allowing for more operation cycles. Thanks to the adapted configurations, the algorithm is able to reach a frequency assignment of the same quality in 88% less time compared to using a unique baseline configuration for all scenarios.

TABLE OF CONTENTS

1. INTRODUCTION

Motivation

Providing broadband internet access through space has been a major target for established and new satellite operators. Iridium, Globalstar, Orbcom, SpaceX, and SES are some example companies that are committed to achieve it thanks in part to the latest advances in satellite technology. The new generation of High Throughout Satellites (HTS) will be able to provide a throughput of hundreds and even thousands of Gbps [1] in order to satisfy the increasing demand from all client segments – especially recent ones, such as maritime or airline transportation.

Flexible satellite payloads have been a key enabler of such unprecedented throughput capacity. By properly reallocating resources, like power or bandwidth, more clients can be served, which increases the combined throughput and ultimately increases revenues [2]. Thanks to technological developments, modern satellite payloads have degrees of freedom orders of magnitude beyond the first flexible payloads. Consequently, resource allocation tasks, which have been traditionally carried out manually or following simple heuristics, have recently undergone a paradigm shift: advanced autonomous systems will be central in the future resource management landscape. In this direction, automatic decision-making able to react to bursty changes in demand is being studied in order to leverage state-of-the-art payload flexibility. What is known as the Dynamic Resource Management (DRM) problem, has been addressed in the recent years with the aim of finding algorithmic solution these autonomous systems can rely on.

Despite the positive results of the solutions proposed in the literature over the last decade, satellite technology keeps evolving fast, and whether previous methods will be able to scale to future systems' requirements has become a new research question. This means that DRM algorithms addressing resource allocation tasks in current satellite systems might fail to do so under new operational conditions. For example, in time-varying scenarios, resource allocation algorithms will need to be executed intermittently in order to adapt to changes in demand or position of the users. If the algorithm is not able to reconfigure and adapt to the new context on time, it might fail to allocate resources optimally, or even worse, operators might incur into service agreement violations.

Taking this worst-case scenario into account, in this paper we focus on the situation in which a certain DRM algorithm is not able to compute the resource allocation within the specific operational time limits because of an inadequate configuration. In other words, algorithms have very limited time to compute how much power or bandwidth to provide to each user and, if they are too slow, operators might fall into competitive disadvantage. Computational speed is largely governed by the algorithm configuration, whether it is the parameters of a convex optimizer or the size of a neural network.

We propose an algorithm-agnostic dynamic reconfiguration methodology that takes into account the scenario addressed by the algorithm – i.e., how many users, which type, how

978-1-6654-3760-8/22/$31.00 ©2022 IEEE

demanding are they, etc. – and configures its hyperparameters accordingly. Based on our results, we discuss that such reconfiguration strategies will be crucial in avoiding fatal operation scenarios and make the most of constellation capacity.

Literature review

The DRM problem in multibeam satellite systems has been broadly studied in the recent years. Different approaches have been proposed, each providing improved solutions for the separate levels of flexibility presented in modern satellite payloads. Generally, these levels of flexibility can be grouped into three main categories: radio-transmitted power per beam, frequency and bandwidth assignment, and beam pointing and beam shaping. This section aims to review previous literature addressing the first two categories, for which NP-hardness has been proven ([3] and [4] for power and frequency, respectively).

The first efforts devoted to the DRM problem were designed in the context of smaller systems present at that time. Satellites like Thaicom 4 [5] encompassed the first algorithms for a dynamic power allocation across its 84 spot beams. Other algorithms proposed back then were designed for satellite systems with similar dimensionality (no more than 100 beams). Authors in [6] did not even consider multibeam coverage when minimizing interference in the frequency assignment problem. In [7] and [8] scenarios with 10 and 20 beams, respectively, were considered for the same problem. Then, power allocation algorithms were tested in similar low-dimensional scenarios: [9] considered 4 beams and [10] considered 10. Such small testing scenarios do not require the algorithms addressing them to be reconfigured. Even if we consider those scenarios as dynamic, i.e. time-varying, algorithms would still be able to address them in real-time given the lower dimensionality.

Other authors considered bigger scenarios and proposed more complex solutions that had to be indeed configured properly. Authors in [3] and [11] used metaheuristics for scenarios with 37 and up to 72 beams, respectively, but did not provide any insight on how the algorithm hyperparameters were decided and reconfigured. The solution proposed in [12] to address a 16-beam scenario borrows its hyperparameters from the work in [13], which optimizes them via Monte Carlo simulation just once. In [14] an algorithm based on Deep Reinforcement Learning was proposed for the frequency plan design of a 37-beam system, but it neither provided any discussion on hyperparameter configuration and how it relates to operation.

Advanced solutions proposed recently with the aim of addressing higher dimensional scenarios also provide little insights on the algorithms' design parameters. In [15] and [16] authors leverage metaheuristics for joint power and bandwidth optimization in 200-beam and 300-beam scenarios, respectively. In [17] a hybrid implementation of Particle Swam Optimization and a Genetic Algorithm is used to address the same problem in a 200-beam system. All of them yield promising results, but only a single hyperparameter configuration is provided despite real-time operations being mentioned as a research thrust. Authors in [18] provide an extensive comparison of algorithms for the power allocation problem in time-varying scenarios, addressing systems with up to 2,000 beams. The effect of scalability is proved to be detrimental for specific metaheuristic algorithms. Moreover, in [19], a similar study is carried out on the applicability of Deep Reinforcement Learning models for the frequency assignment problem. This last work highlights the importance of addressing dimensionality (test cases up to 2,000 beams) and provides significant insights on the relevance of design choices for the proposed solutions, treating state and action representation as hyperparameters whose proper configuration is key to yield robust performances during operation. Still, neither of the papers addresses how algorithm hyperparameters should be reconfigured during scenario changes, including demand and beam variations.

The focus on hyperparameter decisions is widely discussed in other fields that also display certain performance objectives, such as accuracy or quality of the solution, and which also require complex algorithm implementations. In [20] an analysis to configure different algorithms for locomotion learning is presented. Authors in [21] conclude that, for Reinforcement Learning in offline settings, results are not robust to hyperparameter choices and highlight that, when they are controlled carefully, better policies are learned. Even separate algorithmic solutions are proposed in [22] and [23] for hyperparameter optimization in classification with Deep Neural Networks, yielding better results when hyperparameters are optimized.

Taking all the reviewed literature into consideration, we identify a gap when it comes to the hyperparameter adjustment of DRM algorithms. In low dimensional scenarios there might not be a strong requirement for considering adaptive hyperparameter configuration strategies. However, scenarios encompassing more than 1,000 beams – let alone 100 – might not be operable if hyperparameters are not reconfigured accordingly, especially if the scenario requires real-time operation. In systems with considerable number of beams, despite providing discussion and comparisons of different algorithmic implementations, not all papers provide a clear strategy for hyperparameter configuration and reconfiguration. With this in mind, the following section defines the specific objectives this paper tries to cover.

Paper objectives

The specific purpose of the presented work is to develop a strategy to configure the design hyperparameters of algorithms addressing the DRM problem in Satellite Communications by observing the scenario that the algorithms will be facing in real time and adapting its configuration accordingly. To evaluate the performance of the strategy, we use a proprietary Frequency Assignement algorithm as a test case. Our aim is to prove that following a methodology focused on reconfiguration leads to higher benefits in terms of computing time, allowing operators to perform more control cycles without sacrificing performance in the process.

Paper overview

This paper is structured as follows: Section 2 describes the proposed methodology. Section 3 covers the specific test case focused on frequency assignment and the specific implementation of the methodology for it. Section 4 presents the obtained results when following the methodology, and finally Section 5 concludes with the main findings of this paper and its implications in future work.

2. Problem formulation

This section formulates the proposed reconfiguration strategy for the general case, i.e., algorithm-agnostic case. At a given time step of the operation, the DRM system will have to allocate available resources such as power or bandwidth in order to provide the contracted service to users. When allocating

each of the resources, one might look at different features of the scenario to decide how to precisely do it. These features – generally measurable – encode relevant information to assess the complexity of the scenario and might differ depending on the specific resource. For example, if the allocated resource is frequency, some possible features could be the total number of users to serve or their aggregated demand. Those features will be of interest for any DRM algorithm addressing the allocation of frequency.

Given the large amount of time-varying elements present in the system (satellites orbiting, mobile users, bursty demand variations, etc.), scenarios addressed at two consecutive timesteps might not be identical despite their probability of being similar. Given features such as the ones covered in the example belong to a continuous, or highly granular, domain. This makes almost every timestep unique. Consequently, tailoring a specific algorithm configuration for each one of them would not be efficient, since configurations belonging to similar scenarios might look very similar and the slight performance improvement could not justify the overhead reconfiguration costs of the algorithm.

Taking this into account, the focus of the methodology is on **grouping scenarios with similar characteristics, i.e. similar features, and mapping them to discrete levels of "complexity"**. This information can then be used in the process of configuring DRM algorithms for the next allocation cycle by taking the pertinent actions given the complexity of the scenario. This would prevent the algorithm from being constantly reconfigured, which might not be desirable depending on the reconfiguration cost.

Figure 1 depicts how this process would be ideally performed by visually representing it on an hypothetical chart with just two features:

- **Step 1**: In the first step, the features, which have been decided beforehand according to the allocated resource, are computed and represented. For example, each data point could represent a different beam distribution.
- **Step 2**: Next, scenarios with similar features are grouped into clusters by means of a clustering algorithm.
- **Step 3**: Different hyperparameter configurations are then assigned to each cluster so that viability of operation is granted and algorithm resources are leveraged at its best. This step depends on the specific DRM algorithm we want to wrap a hyperparameter reconfiguration policy around, since hyperparameter specifics might differ depending on the type of algorithm.

We identify two main advantages of this methodology, in contrast to fixing the algorithm configuration for all possible scenarios:

- Scenarios belonging to low complexity clusters would benefit from algorithm configurations that allow more intense computation. For example, iterative algorithms could afford a higher number of iterations and reach more optimal solutions. Constraint-based algorithms would face easier search spaces in this case, leading to a higher chance of finding the global optimum in the given time window.
- Viability of operation under complex scenarios would be granted by assigning prudent values to the hyperparameters of the algorithm and, therefore, limiting the size of the search space and reducing the time required to complete an allocation cycle. This time versus optimality tradeoff would prevent the algorithm from exceeding the time window imposed by real time operation while still providing a valid solution for the resource allocation.

3. Paper use case: Frequency assignment

This section covers, first, the specific resource allocation problem and the proprietary algorithm being configured to test the methodology; and second, a detailed view of the specific implementation of the methodology for this case.

The resource allocation algorithm

The test case DRM algorithm considered in this paper is a frequency assignment algorithm; the methodology described in the previous section is applied to it. As explained, instead of pre-fixing its hyperparameters and then testing them on multiple scenarios, the methodology we propose does this process in the opposite direction, that is, first observe the scenario and then configure the algorithm properly, so that viability of operation is granted. This is strictly necessary if the frequency allocation algorithm is expected to operate in time-varying scenarios, where the position and demand of the users is not static and the optimization of resources has to be rethought at each time step to maximize efficiency. In this situation, if the algorithm's configuration is fixed and the scenario at a certain time step is not operable with that configuration, the algorithm might not be able to complete the frequency allocation and therefore fail to provide users with the contracted service.

In order to avoid this situation, we identify the need of observing and objectively characterizing the scenario of the frequency allocation problem before configuring and executing the algorithm addressing it. An schematic of the scenario is depicted in Figure 2. We assume that at this stage of the resource allocation process, the users have already been grouped into beams and routed to gateways through the satellites of a Non-Geostationary (NGSO) constellation. Then, the purpose of the algorithm is to assign a part of the frequency spectrum to each of the beams in order to satisfy the demand of the users covered by those beams. Since we are dealing with NGSO systems, handovers between satellites might occur and those should not hinder operations. A handover corresponds to the process by which a beam changes its routing from one satellite to another that has better coverage. Generally, beams point to fixed locations, so the satellite powering them "switches" over time. The work in this paper is focused on the frequency assignment for downlink beams. In the following sections, problem scenario and beam distribution might be used interchangeably.

Generally speaking, algorithms addressing the frequency allocation problem are focused on providing a feasible frequency assignment for a certain beam distribution.Despite algorithm differences, all methods are addressing the same problem, the same decisions, and the same variables. These variables completely define the frequency plan and can be grouped into four categories:

- *Bandwidth*: Variables referring to the amount of bandwidth assigned to each beam. They can be continuous or discrete, considering multiples of a minimum unit of bandwidth.
- *Center frequency*: Variables referring to the position of the bandwidth interval assigned to each beam in the available spectrum.

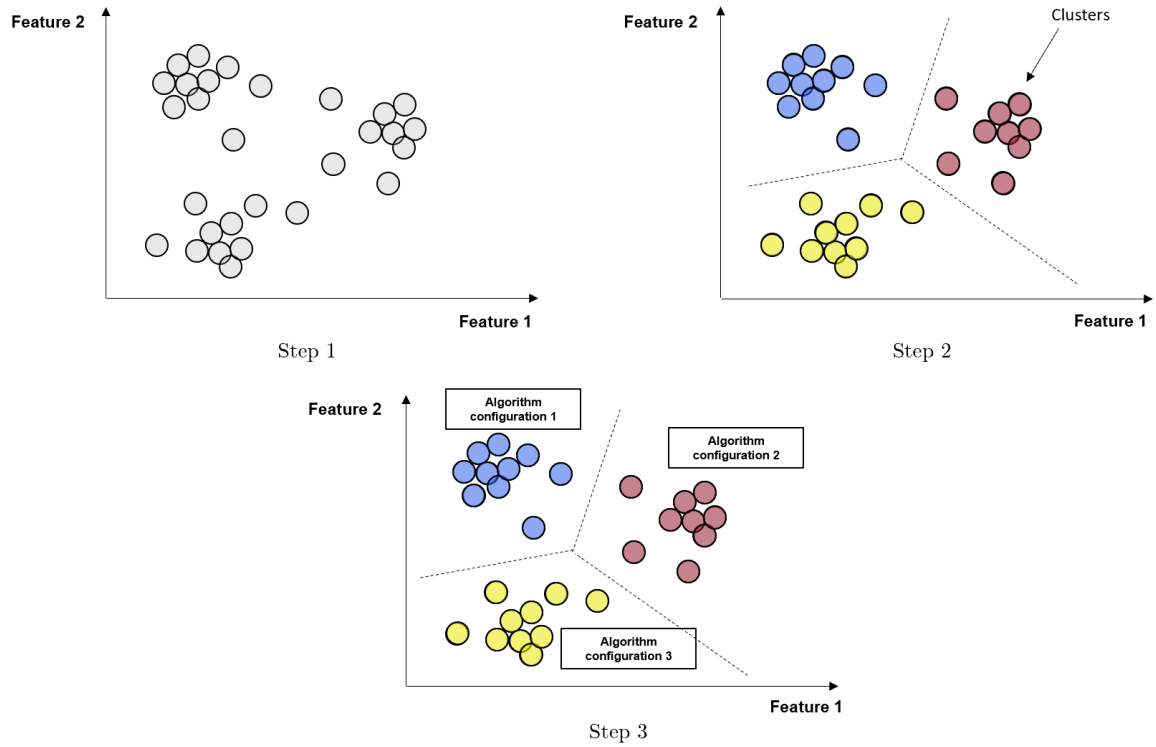


Figure 1: Schematic of the methodology.

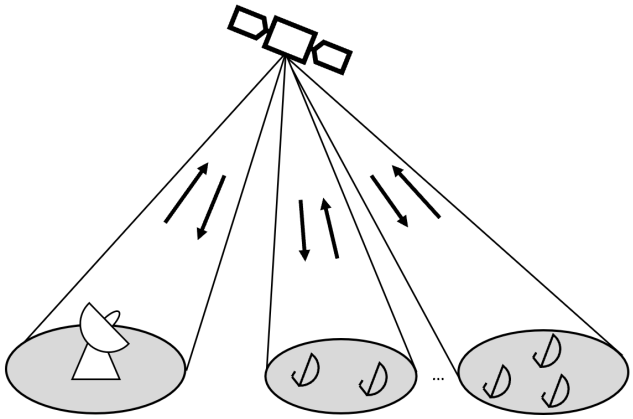


Figure 2: Scenario addressed by a frequency plan optimization algorithm before operation.

• *Polarization*: Variables indicating the polarization used for each beam.

• *Reuse group*: Some satellite systems allow for frequency reuse by means of reuse groups. We assume they are available in our test case too. Then, certain variables, one per beam, encode different reuse groups.

A frequency plan is completely defined when each of the beams of the constellation has an amount of bandwidth, a center frequency, a polarization, and a reuse group assigned to it.

When assigning the aforementioned variables, certain limitations need to be taken into account. We formally denote these limitations as restrictions of the problem and, when it comes to frequency assignment, they can be grouped into two main categories:

• *Interference restrictions*: If two beams, whose footprints are close, are powered by the same satellite and use the same polarization, they might interfere with each other if their assigned bandwidths overlap. This imposes an interference restriction between them, which should be respected by the optimization algorithm. The criteria upon which the interference restrictions are set is flexible: in our test case algorithm a threshold angular separation is used, but other criteria involving *SINR* metrics might also be applied.

• *Handover restrictions*: In the event of a handover, a beam that is handed over to a new satellite must not overlap in frequency with the ones already being served by that satellite, if they use the same reuse group and polarization. Therefore, two beams that will be served by the same satellite at some point during the time window considered must have a bandwidth limitation with the beams in the same frequency group (reuse group and polarization). We call this limitation a *handover restriction*.

The restrictions of the problem reduce the space of valid configurations, i.e. the amount of possible solutions to the frequency assignment problem. Complicating the search space of a problem directly affects the computing time performance of any algorithm trying to solve it, and its ability to find the global optimum. We leverage this property and use the number of restrictions as a measure of the difficulty of any algorithm to find a solution or find it fast enough.

The test case algorithm is a proprietary algorithm based on Mathematical Programming aiming to optimize the fre-

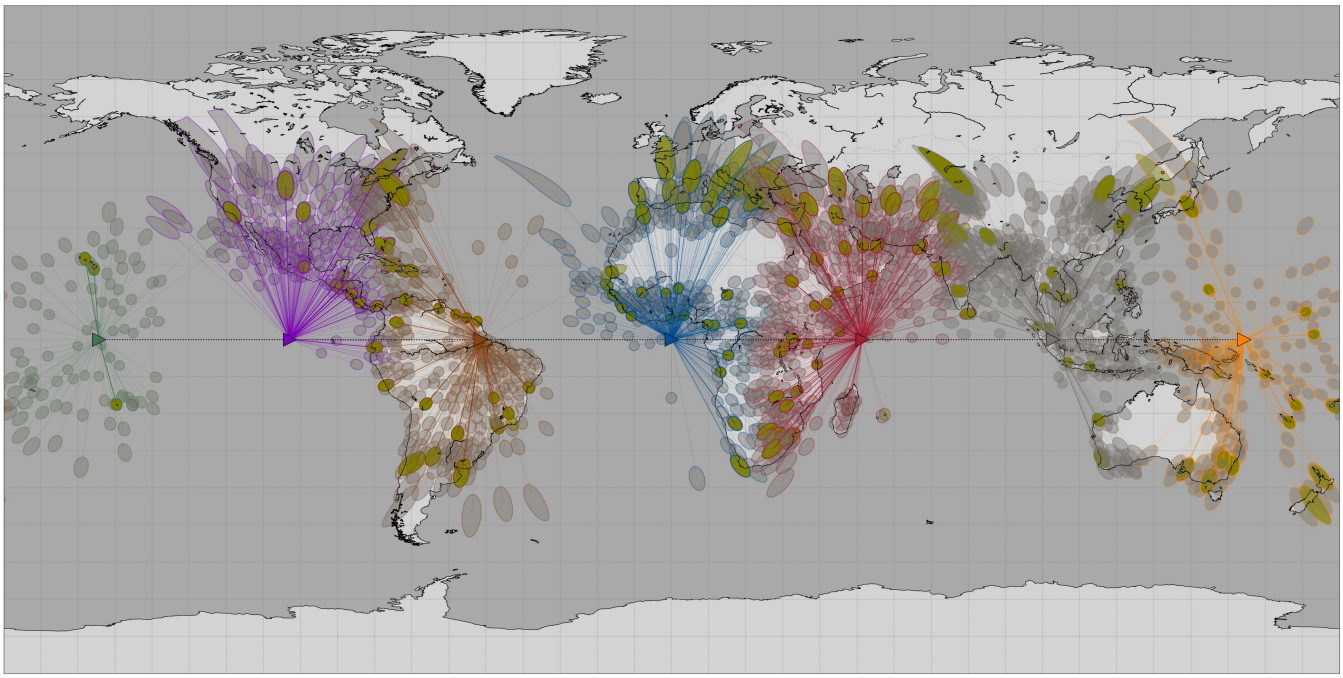


Figure 3: Beam distribution used for training the model.

quency plan of an NGSO Satellite Constellation in real time. This algorithm follows an iterative process in order to improve an already viable but suboptimal frequency plan used as a warm start. Typically, Mathematical Programming algorithms directly provide the optimal solution without iteration or convergence processes. However, this is not possible in high-dimensional scenarios as those imposed by the upcoming satellite communications landscape. In this case, if computing time limitations are taken into account, the algorithm is not able to provide a solution within the required time without iterating, as it does not scale linearly.

For each iteration, the algorithm selects a subset of beams and optimizes the frequency allocation for those beams while keeping the allocation of the rest of the beams fixed in the form of constraints. This process is repeated over time, selecting different beams at each iteration and stopping when the number of iterations limit is reached or the solution has converged, whichever comes first. The hyperparameters that govern the algorithm are:

- **Total number of beams**: This number represents the number of beams in the constellation, which might range from hundred to thousands depending on the specific constellation considered.
- **Number of beams optimized at each iteration**: This is the most relevant hyperparameter regarding the algorithm performance. Beams are optimized in a iteration-based fashion. The more beams optimized together at once, the closer the intermediate solution is to the global optimum. However, since the algorithm does not scale linearly, adding more beams per iteration negatively impacts computing time.
- **Bandwidth optimization flag**: Binary variable indicating whether the amount of bandwidth assigned to each beam can be changed or it is fixed to the warm start amount. Adding more bandwidth is generally beneficial, since the more bandwidth used, the less power needed to provide the required data rate. However, allowing bandwidth changes can substantially increase the size of the search space, and therefore impact computing time. Therefore, there might be situations where fixing the bandwidth might be required in order to satisfy computing time limitations.
- **Number of maximum iterations**: It is important to set this hyperparameter taking into account the overall available computing time.

Setting these hyperparameters properly will define the viability of the optimization process, and developing a methodology for doing so has been the specific objective of the work in this paper.

Implementation of the methodology in the frequency assignment use case

This section provides the specific details of the implementation of the methodology introduced in 2. We mainly discuss the specific features, the clustering algorithm used and its training process, and the metrics used to measure the improvement granted by the methodology.

The first step of the implementation of the methodology described in the previous section consists of selecting the appropriate features that better describe a scenario, in our case a beam distribution. To that end, the features measured for each beam distribution are the following:

- **Number of beams**, f_1: The overall number of beams in a beam distribution is one of the main informative features of the beam distribution.
- **Average demand per beam**, f_2: This feature captures the demand of each of the beams in the distribution by adding up the data rate contracted by the users in each beam and averaging across all of them. Its units matches the units of demand or data rate, e.g. Mbps. The aim of this feature is to provide a measure that is proportional to the amount of bandwidth required by the beams on average (higher demand

per beam will require a higher bandwidth assigned to the beams, on average).

- **Average number of interference restrictions per beam**, f_3: The purpose of this feature is to measure how restricted the problem search space is. As explained in section 3, the restrictions of the search space are directly proportional to the difficulty for an algorithm to explore it.

- **Average number of handover restrictions per beam**, f_4: Similar to the previous one, this feature provides information on how restricted the search space is by looking at the other type of restrictions. Based on our experiments, the average number of handover restrictions is around an order of magnitude above the number of interference restrictions. This gap decreases when there are more dense areas in the scenario.

In order to improve generalization, we also create additional features based on transformations of the already existing ones. With that purpose, in addition to the proposed features, the following transformations are also applied (f and t denote the initial feature and the transformed feature, respectively):

- Power of 2: $t_{i+3} = f_i^2$, $\forall i \in \{2, 3, 4\}$
- Logarithm: $t_{i+6} = \ln(1 + f_i)$, $\forall i \in \{2, 3, 4\}$
- Inner product: $t_{11} = f_1 \cdot f_2 \cdot f_3$ and $t_{12} = f_1 \cdot f_2 \cdot f_4$

Taking the transformations into account, we end up with a total of 12 features. More features would be created, but what matters in this process is ending up with the fewest number of features that captures as much variability as possible. To that end, before clustering, we apply *Principal Component Analysis* (PCA) in order to decompose those 12 features into a reduced number of uncorrelated features (or principal components) that preserve the largest part of the variation. In our case, we reduce the feature space to a 2-dimensional space for visualization purposes, but a larger number of principal components could also be employed.

Next, we cluster the obtained principal components using the *k-means clustering* algorithm. We use this specific algorithm because of its inherent simplicity and quick convergence time compared to other clustering algorithms. In our experiments with thousands of beams we found out it took less than a second to run the algorithm, making it an almost-zero overhead cost when executed before one optimization run. To decide a good number of clusters, we select the value that yields the highest *Silhouett Score* based on the available data.

Typically, the more data available to train the clustering model, the better its generalization to unseen data will be. As explained in section 2, each data point fed to the model corresponds to a different beam distribution. Therefore, we would like to train our clustering model with a dataset containing as many different beam distributions as possible.

However, many operators will only have access to single user pools consisting on their current or potential future clients. For this reason, we leveraged a single-distribution dataset to generate the data to train the clustering algorithm.

In our case, we work with data provided by SES from 20,000 users scattered all over the world, which results into a beam distribution of approximately 5,000 beams [24]. The resulting beam distribution is depicted in Figure 3. This beam distribution is an example scenario that could be faced by a Frequency Assignment algorithm. It can be seen that users have already been grouped into beams and beams have been routed to the corresponding satellite covering them. The satellites, represented with a triangle, belong to a seven-satellite MEO constellation.

In order to generate the training data from this single user distribution we follow a partitioning strategy. Concretely, we divide the beam distribution into cells of different size (1ºx1º, 5ºx5º and 10ºx10º) with 1º granularity in each direction (N-S, E-W) and treat each cell as an independent beam distribution. Figure 4 illustrates this approach schematically.

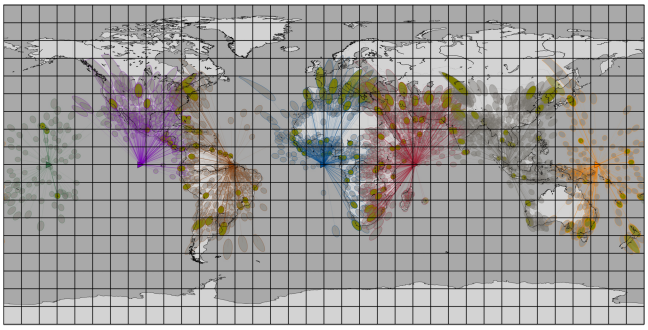


Figure 4: Schematic of the partition strategy followed in order to leverage a single beam distribution.

We use this approach for the training stage using each cell as a separate data point. Then, the metrics we use for measuring the performance improvement granted by the methodology objectively are:

- **Average computing time per iteration**: Average of the time needed to complete one iteration across all the iterations in one optimization run.

- **Maximum computing time per iteration**: Computing time required to complete the longest iteration of the optimization. Since each iteration optimizes the frequency plan for different subsets of beams, with different number and distribution of restrictions each, the computing time per iteration is not constant.

- **Bandwidth increase**: Normalized increment of the overall bandwidth used by all beams during an optimization run. As explained in Section 3, there exists a tradeoff between increasing the bandwidth of each beam and the computing time required to complete the optimization. This tradeoff is reflected on this metric. If the bandwidth optimization flag is set to 0, the beams will have the same fixed amount of bandwidth during the entire optimization run and, therefore, there will be no bandwidth increase.

- **Number of iterations until convergence**: Number of iterations required to reach a stable solution of the frequency allocation of the beams.

- **Convergence time**: Overall convergence time, which can also be understood as the sum of the computing time of each iteration until convergence is reached.

Since reaching convergence might not always possible due to the scenario dimensionality, the last two metrics are computed in a smaller set of the testing cells for which the algorithm is allowed to reach convergence. By setting a higher iteration limit, the overall bandwidth used by the beams reaches an stable value. This way, we can draw conclusions on the benefit of the methodology when it comes to the convergence of the solution.

4. Results

In this section we introduce the results obtained from clustering the generated training dataset and validate them by plotting them back on the beam distribution. Finally, we report the improvements granted by adapting the hypeparameters to each of the identified clusters.

Clustering results

After computing the specified features for each cell containing at least one beam (empty cells are excluded) and applying the Principal Component Analysis on the data, we decide the number of clusters for the k-means algorithm with the help of the Silhouette Score introduced in the previous section. The Silhouette Score measures the quality of the obtained clusters based on the data distribution and the number of clusters – i.e., parameter k. Computing it for different number of clusters allows us to use the number of clusters that yields a better Silhouette Score value. Figure 5 shows this plot for our case.

Figure 5: Silhouette score for different number of clusters.

From the image, the highest Silhouette Score is obtained with three clusters, therefore we use $k = 3$ to cluster the data. A close value is obtained with 5 clusters, which could be a reasonable choice too given the structure of the dataset. In that sense, all the analyses to come have been replicated for $k = 5$ in the Appendix A. Once the number of clusters is decided, we run the k-means clustering algorithm and obtain the results shown in Figure 6.

The principal components conform an orthogonal basis that captures the maximum amount of variance in the data but has no explainable meaning for our specific context. To get a more interpretable view of the solution, we map the clusters back to the initial features and plot the different clusters with meaningful axes. Figures 7 and 8 show this mapping for the normalized average demand per beam vs. the number of interference restrictions per beam and vs. the number of handover restrictions per beam, respectively.

From these results, we identify three clusters that can be mapped to three different areas of "complexity":

- **Cluster Easy**: Areas with low demand per beam and few restrictions between beams. Search space can be explored with ease.
- **Cluster Intermediate**: Areas with intermediate values for demand and interference restrictions per beam, but with large numbers of handover restrictions between beams. However, handover restrictions are less restrictive than interference restrictions and therefore the search space is not too complex.
- **Cluster Hard**: Areas with the highest demand per beam and largest number of restrictions per beam (both types). The search space is highly restricted and therefore exploration is difficult.

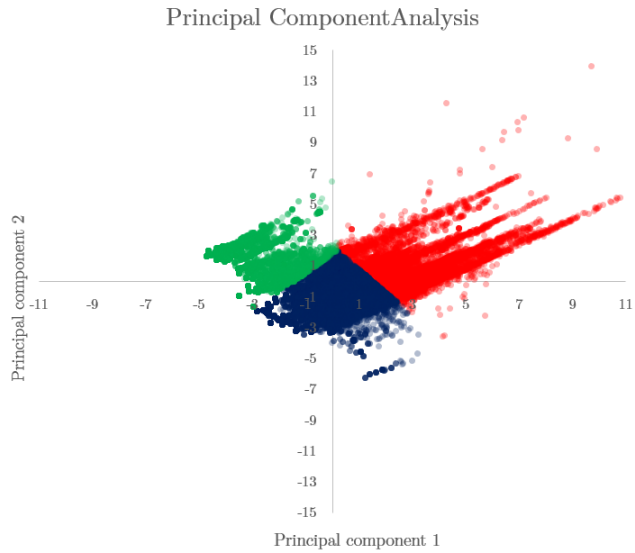


Figure 6: Clustered data after Principal Component Analysis.

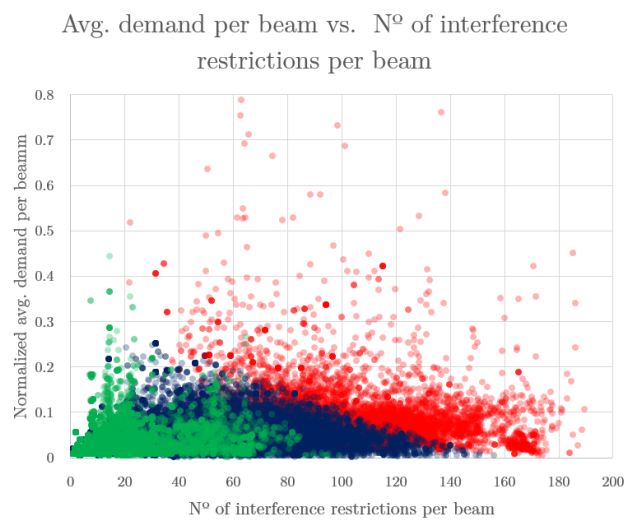


Figure 7: Normalized average demand per beam vs. number of interference restrictions per beam.

Figure 9 shows the clustering results plotted onto the beam distribution used for training. Specifically, the cells used in the training distribution are plotted and colored onto the map shown in Figure 3 (only 10ºx10º cells are plotted for visualisation purposes).

Based on these results, we can focus on different areas of the globe and discuss the coherence of the obtained clusters:

- Users in the Pacific area consist mostly of highly-separated maritime terminals, e.g. leisure cruises, dispersed all over

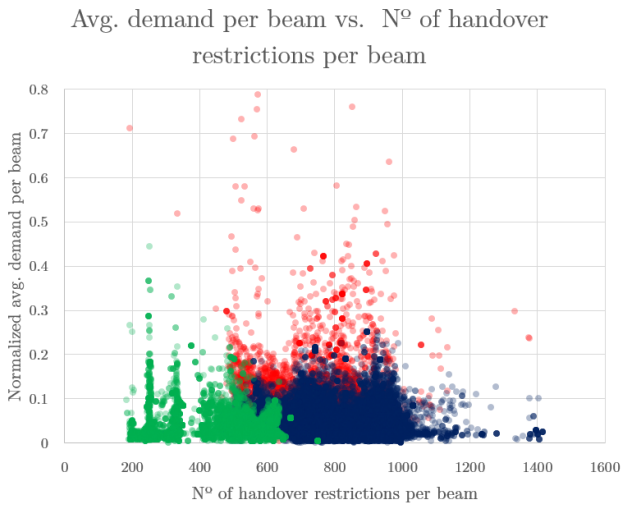


Figure 8: Normalized average demand per beam vs. number of handover restrictions per beam.

the sea. This implies that most of the time they will require a separate beam and the overall distribution will be sparse. This results in a smaller average number of restrictions among beams, since they are far from each other. Moreover, the impossibility of grouping users in the same beam results in beams with lower demand. That is why in Figure 9 we see the vast majority of the pacific ocean covered in green.

- The Atlantic and Indian oceans are home to strategic commercial routes, and therefore cruise traffic is denser. This causes beams to be closer to each other compared to the Pacific area and even more than one terminal might be grouped into the same beam if possible, which translates into an intermediate (blue color) area of optimization complexity.

- Similarly, land areas like Africa, northern South America or Australia are also covered in blue but, in this case, this is due to a higher user density compared to the oceans. However, since a fewer number of user terminals are present in those areas, this results in an intermediate complexity, i.e., optimizing is neither too easy nor too hard.

- Finally, areas like north America, Europe and central Asia concentrate the majority of users, resulting into highly demanding beams very close to each other. This is why these areas are the hardest to optimize and are mapped to the hard cluster (red color).

Algorithm Performance results

This section presents the results obtained from applying the methodology to the test case algorithm and analyzes how the algorithm behaves when trying to optimize beam distributions belonging to each of the three different identified clusters.

Before presenting the results for the different scenario complexity levels, we present the configurations that have been tested in each of them, as well as the baseline configuration that will serve as reference. Each configuration consists of a set of specific values for the algorithm parameters presented in Section 3. Table 1 presents a summary of the selected configurations: the Baseline configuration used as reference and the Adapted configuration for each complexity level. As a baseline, we used a suboptimal configuration that was previously used in our research when the beam distribution complexity or the computing time limitations were not taken into account.

Configurations				
	Baseline	Adapted		
		Easy	Intermediate	Hard
B	*	*	*	*
C	20	15	10	5
Bw	1	1	1	0
$\mathbf{N_{iter}}$	20	20	20	20

Table 1: Algorithm configurations used to test the methodology (*Number of beams automatically set to the number of beams inside each cell). **B**: Size of the pool of beams to optimize. **C**: Number of beams optimized per iteration. **Bw**: Adjustable bandwidth, 1 if enabled, 0 if not. $\mathbf{N_{iter}}$: Number of iterations

On the other hand, we used different configurations when taking the scenario complexity into account, resulting into different Adapted configurations for each complexity level. For each complexity level, we gradually reduce the number of beams optimized per iteration and fix the bandwidth for the hardest scenarios. These are example configurations that have been adjusted based on our specific dataset. A different set of configurations might be necessary when using a different constellation as a reference model.

As a first test, for each complexity level, we perform a 20-iteration optimization run two times for each cell: one with the baseline configuration and one with the adapted configuration. We perform this test for a random set of 120 cells of 10ºx10º (40 of each complexity level) and compare the results between the baseline and adapted configurations. Figure 10 shows the histograms of each of the three first metrics introduced in Section 3, with the dashed lines representing the average for each complexity level. We distinguishing between levels by colors (Easy, Intermediate and Hard). Table 2 summarizes the average results.

For the first complexity level, corresponding to the cluster Easy, there is no noticeable improvement with respect to the baseline configuration. This is due to the fact that those cells contain a very small amount of beams, making the iteration-based procedure almost unnecessary. When this happens, the number of beams per iteration (C) is automatically adjusted to the number of beams inside the cell (B), resulting in the same configuration for the baseline and adapted cases, and therefore no improvements are appreciated.

On the other hand, results are different for the intermediate and hard complexity levels (clusters Intermediate and Hard, respectively). In both cases, there is a considerable reduction of the average computing time per iteration of 50% and 79%, respectively. This reduction is also reflected in the maximum computing time per iteration, with 69% and 88%, respectively. When it comes to the bandwidth increase, however, a 5% increment is noticed in the intermediate complexity level whereas a 100% reduction is observed for the hard. Due to the fact that the adapted configuration for the hard scenarios has its bandwidth flag (Bw) set to 0, there is no bandwidth increase for that specific optimization run. Fixing the bandwidth of each beam, however, results in the appreciated reduction in computing times.

As mentioned, in order to analyze the results from a convergence perspective, we select a smaller set of 15 cells (5 for each complexity level) and perform the same tests, this time

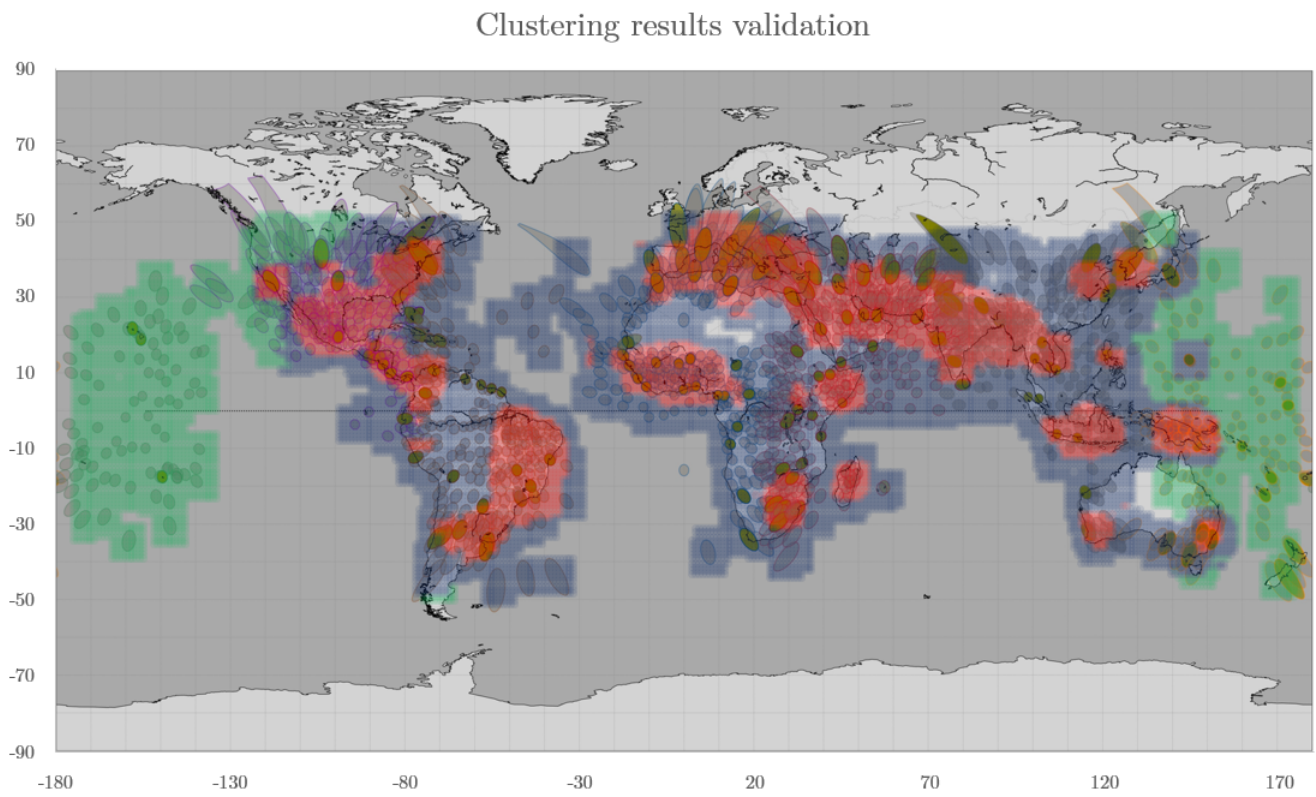


Figure 9: Clustering results of 10ºx10º cells plotted above the beam distribution.

		Average results		
		Avg. comp. time (s)	**Max. comp. time (s)**	**Bw. increase**
Easy	Baseline	0.07	0.08	9.0
	Adapted	0.07	0.08	9.0
	Δ (%)	**0.4**	**0.8**	**0.0**
Interm.	Baseline	0.25	0.54	7.3
	Adapted	0.12	0.17	7.7
	Δ (%)	**-50**	**-69**	**5**
Hard	Baseline	0.85	1.93	3.0
	Adapted	0.18	0.24	0.0
	Δ (%)	**-79**	**-88**	**-100**

Table 2: Average results of the test with 120 cells.

allowing the algorithm to reach convergence by increasing the number of iterations to larger values and stopping the optimization run once a stable solution is reached. The results for the specific metrics are reported in Figure 11 and Table 3.

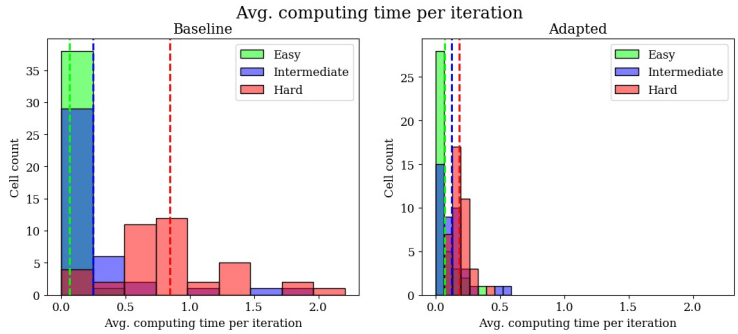
Similar to the previous results, low complexity cells do not seem to benefit from adapting the configuration of the algorithm, with a negligible improvement in the number of iterations until convergence and even a slight increase of 1.1% in the convergence time.

Again, however, positive results are obtained from the intermediate and hard cells. In the first case, despite the 43% increase in the number of iterations required to converge, a 73% reduction in the overall convergence time is noticed. This is due to the fact that the intermediate configuration reduces the number of optimized beams per iteration from 20 to 10 beams, allowing the algorithm to run faster for each iteration and reach convergence sooner.

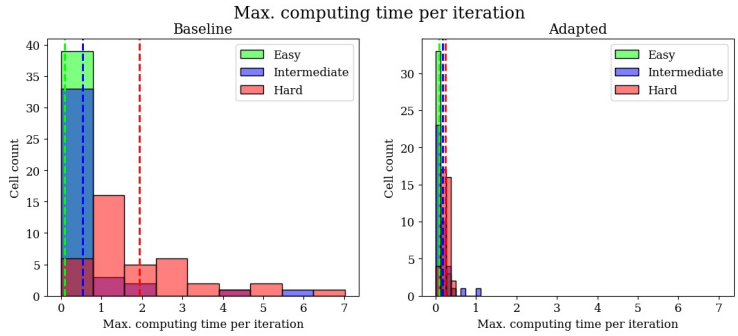
In the second situation, for hard complexity cells, there is a reduction both in the number of iterations required to converge and the overall convergence time, of 6.6% and 88%, respectively. The rationale behind this numbers is, again, found in the adapted configuration for the hard scenarios. The number of optimized beams per iterations is reduced to 5 beams and the bandwidth is kept fixed, allowing the algorithm to speed up each iteration and reach an stable solution within a much reduced computational time.

5. Conclusions and Future Work

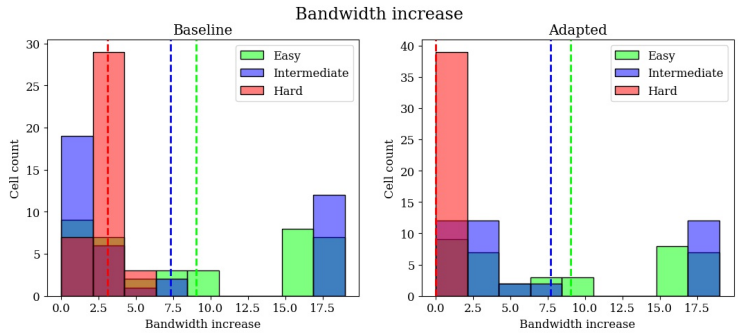
In this paper we have presented a strategy to reconfigure the design hyperparameters of algorithms addressing Dynamic Resource Management (DRM) problems in Satellite Communications. Assuming real-time and high-dimensional contexts, our method "observes" the scenario the algorithm must address in real time and adapts its configuration accordingly.



(a) Average computing time per iteration.



(b) Maximum computing time per iteration.



(c) Bandwidth increase (Total bandwidth used by the optimized beams after the optimization).

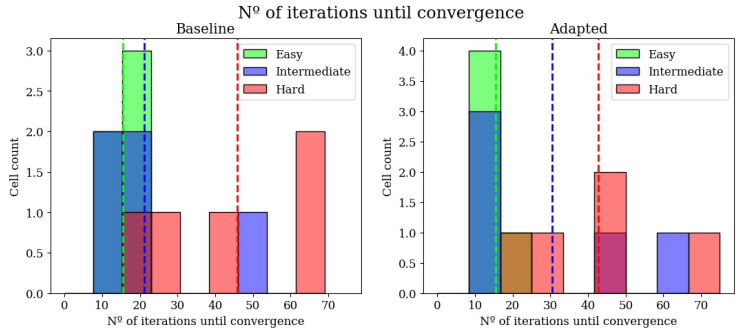
Figure 10: Results for 120 testing cells (40 of each complexity level). For each complexity level, two tests per cell have been carried out: one with the baseline configuration and one with the adapted configuration of the corresponding level.

First, we formulated the problem by describing the proposed methodology. Next, we described the test case algorithm used to demonstrate the usefulness of our method, providing some insights on the frequency assignment problem in the process. Finally, we presented the results of applying our methodology to the frequency assignment algorithm. The main conclusions extracted from these results are the following:

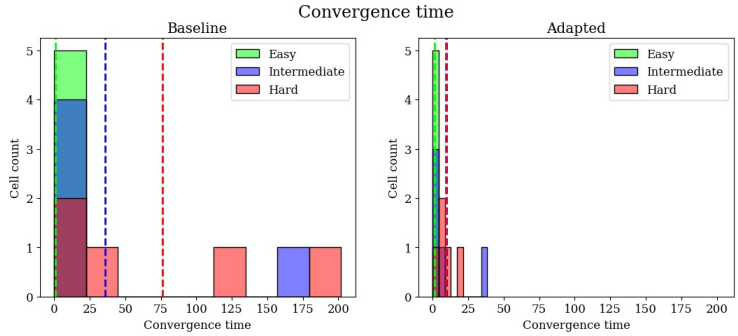
- Observing the scenario addressed by the algorithm and its features before reconfiguration is a key step. In this paper we identified clear differences of scenario complexity when looking at different areas of the beam distribution. These differences are present not only spatially but also temporally, with highly different beam distributions when considering different moments of the day, week, month or year. Therefore, and especially when computing resources are scarce or scenario dimensionality is high (e.g., thousands of beams), a complete study of the scenario complexity is crucial before configuring the algorithm addressing it.

- When differences between plausible scenarios are identified, the best way to address them is by tailoring different algorithm configurations for each of them. This is a process that has not been addressed in the DRM algorithm literature so far. In this paper, we have obtained tangible improvements by doing so, i.e., adapting the algorithm configuration to each identified type of scenario instead of using the same configuration for all scenarios. Benefits are clear: the adapted configurations where able to reach a 79% reduction in computing time per iteration in the best case, and a 88% reduction in the overall convergence time. Moreover, for some scenarios, computing time per iteration was a 50% lower while even increasing the overall bandwidth used by 5%. In the light of these results, we believe there is a clear advantage of adapting the algorithm configuration based on scenario observation and encourage the use of this methodology in DRM studies moving forward.

Possible extensions of the work presented in this paper are:



(a) Number of iterations until convergence of the solution.



(b) Time required to reach an stable solution or converge.

Figure 11: Convergence results of 15 cells (5 of each complexity level).

Convergence results			
		Num. of iterations until convergence	**Convergence time (s)**
Easy	Baseline	15.4	1.0
	Adapted	15.4	1.0
	△ (%)	**0.0**	**1.1**
Interm.	Baseline	21.2	36.0
	Adapted	30.4	9.62
	△ (%)	**43**	**-73**
Hard	Baseline	45.8	76.3
	Adapted	42.8	8.94
	△ (%)	**-6.6**	**-88**

Table 3: Average results of the test with 15 cells until convergence.

- Apply the same or a similar methodologies to configure other already-existing algorithms addressing the DRM problem in real-time and high-dimensional scenarios.
- For future or currently in development DRM algorithms, implement a scenario identification and algorithm configuration stage before actually addressing the specific resource allocation problem.
- Explore other elements that could be incorporated in the presented methodology: different clustering algorithms, other supervised learning methods, and trying different target metrics in order to focus on different objectives other than computing time.
- Finally, explore other domains where high-dimensional and time-varying scenarios are considered, e.g. operations research, and apply a similar methodology for problems in the field.

APPENDICES

A. SCENARIO IDENTIFICATION RESULTS WITH K=5 CLUSTERS

The Silhouette Score obtained when setting the number of cluster $k = 5$ was close to the highest value, obtained with $k = 3$, as shown in Figure 5. In this section we include the replicated results of this paper when using $k = 5$ clusters. It can be seen that the algorithm clusters the data similarly than in 4, but with two additional clusters.

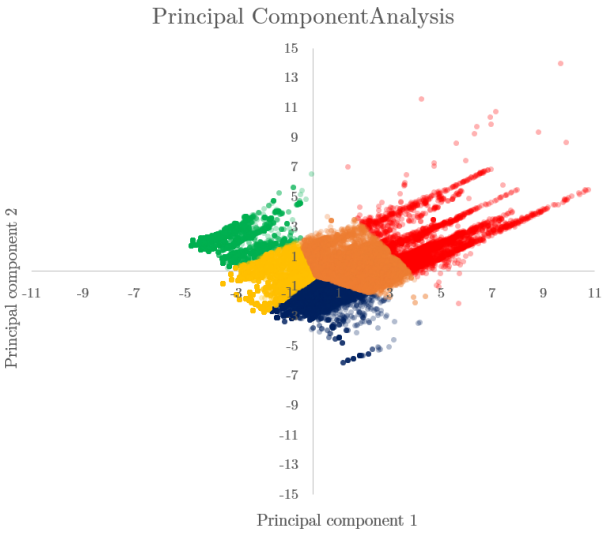


Figure 12: Clustered data after Principal Component Analysis

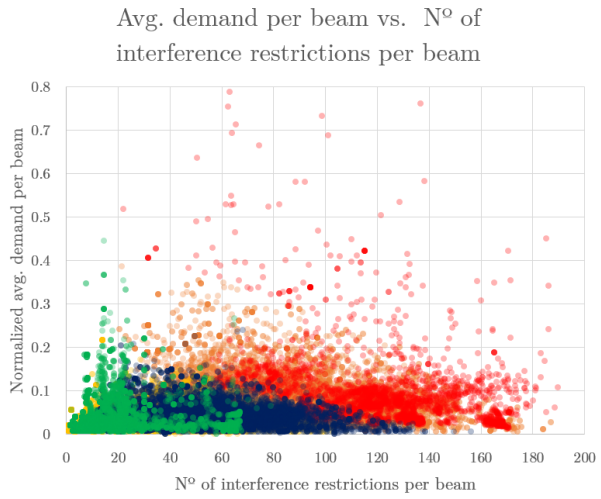


Figure 13: Avg. demand per beam vs. number of interference restrictions per beam

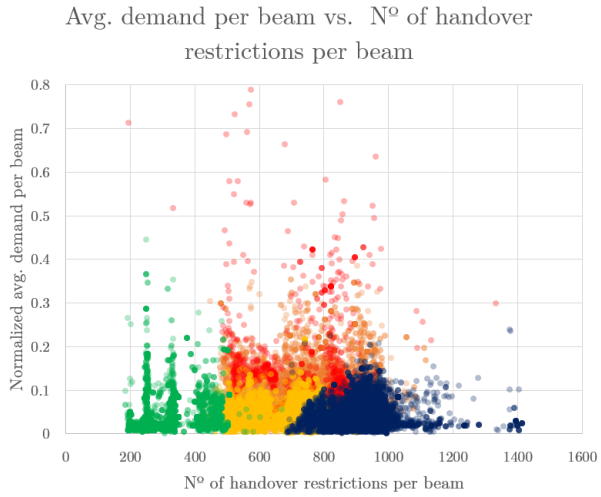


Figure 14: Avg. demand per beam vs. number of handover restrictions per beam

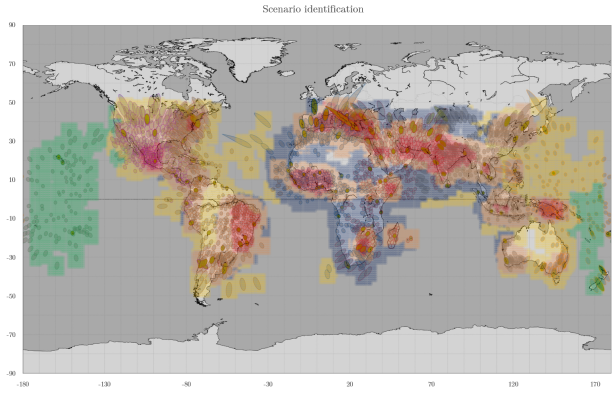


Figure 15: Clustering results of 10ºx10º cells plotted above the beam distribution

Acknowledgments

This work was supported by SES. The authors want to thank SES for their input to this paper and their financial support.

References

[1] SES, "O3b mPOWER," https://www.ses.com/our-coverage/o3b-mpower, May 2021. [Online]. Available: https://www.ses.com/our-coverage/o3b-mpower

[2] M. Guerster, J. J. G. Luis, E. F. Crawley, and B. G. Cameron, "Problem representation of dynamic resource allocation for flexible high throughput satellites," in *2019 IEEE Aerospace Conference*, 2019.

[3] 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, 2015.

[4] T. Mizuike and Y. Ito, "Optimization of frequency assignment," *IEEE Transactions on Communications*, vol. 37, no. 10, pp. 1031–1041, 1989.

[5] Gunter, "iPStar 1 (Thaicom 4, MEASAT 5, Synertone 1) - Gunter's Space Page," aug 2005. [Online]. Available: https://space.skyrocket.de/doc_sdat/ipstar-1.htm

[6] S. Salcedo-Sanz and C. Bousoño-Calzón, "A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications," *Applied Intelligence*, vol. 22, no. 3, pp. 207–217, May 2005. [Online]. Available: https://doi.org/10.1007/s10791-005-6619-y

[7] Heng Wang, Aijun Liu, Xiaofei Pan, and Luliang Jia, "Optimal bandwidth allocation for multi-spot-beam satellite communication systems," in *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, 2013, pp. 2794–2798.

[8] U. Park, H. W. Kim, D. S. Oh, and B.-J. Ku, "A dynamic bandwidth allocation scheme for a multi-spot-beam satellite system," *ETRI Journal*, vol. 34, no. 4, pp. 613–616, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.12.0211.0437

[9] H. Wang, A. Liu, X. Pan, and J. Yang, "Optimization of power allocation for multiusers in multi-spot-beam satellite communication systems," *Mathematical Problems in Engineering*, vol. 2014, p. 780823, Mar 2014. [Online]. Available: https://doi.org/10.1155/2014/780823

[10] Y. Hong, A. Srinivasan, B. Cheng, L. Hartman, and P. Andreadis, "Optimal power allocation for multiple beam satellite systems," 02 2008, pp. 823 – 826.

[11] A. Paris, I. Del Portillo, B. Cameron, and E. Crawley, "A genetic algorithm for joint power and bandwidth allocation in multibeam satellite systems," in *2019 IEEE Aerospace Conference*, 2019, pp. 1–15.

[12] F. R. Durand and T. Abrão, "Power allocation in multibeam satellites based on particle swarm optimization," *AEU - International Journal of Electronics and Communications*, vol. 78, pp. 124–133, 2017.

[13] T. Abrao, L. D. H. Sampaio, M. Jr, B. Angelico, and P. Jeszensky, *Multiple Access Network Optimization Aspects via Swarm Search Algorithms*, 04 2011.

[14] X. Hu, S. Liu, R. Chen, W. Wang, and C. Wang,

"A deep reinforcement learning-based framework for dynamic resource allocation in multibeam satellite systems," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1612–1615, 2018.

[15] 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, 2018.

[16] J. Lei and M. A. Vázquez-Castro, "Joint power and carrier allocation for the multibeam satellite downlink with individual sinr constraints," in *2010 IEEE International Conference on Communications*, 2010, pp. 1–5.

[17] N. Pachler, J. J. G. Luis, M. Guerster, E. F. Crawley, and B. G. Cameron, "Allocating power and bandwidth in multibeam satellite systems using particle swarm optimization," in *2020 IEEE Aerospace Conference*, 2020.

[18] 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.

[19] J. J. G. Luis, E. F. Crawley, and B. G. Cameron, "Applicability and challenges of deep reinforcement learning for satellite frequency plan design," in *2021 IEEE Aerospace Conference*, 2021.

[20] D. Reda, T. Tao, and M. van de Panne, "Learning to locomote: Understanding how environment design matters for deep reinforcement learning," *Motion, Interaction and Games*, Oct 2020. [Online]. Available: http://dx.doi.org/10.1145/3424636.3426907

[21] T. L. Paine, C. Paduraru, A. Michi, C. Gulcehre, K. Zolna, A. Novikov, Z. Wang, and N. de Freitas, "Hyperparameter selection for offline reinforcement learning," 2020.

[22] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-weka: Combined selection and hyperparameter optimization of classification algorithms," 2013.

[23] P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor, "Particle swarm optimization for hyperparameter selection in deep neural networks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 481–488.

[24] N. Pachler, M. Guerster, I. del Portillo, E. F. Crawley, and B. G. Cameron, "Static beam placement and frequency plan algorithms for leo constellations," *International Journal of Satellite Communications and Networking*, 2020.

BIOGRAPHY

Sergi Aliaga *is a MSc student in the Ultra-Boradband Nanonetworking (UN) laboratory of the Electrical and Computer Engineering Department at Northeastern University. He received two BSc in Telecommunications Engineering and Industrial Engineering from the Polytechnic University of Catalonia, under the CFIS (Interdisciplinary Higher Education Centre) program. Sergi carried out his Bacherlor Thesis at MIT on reconfiguration strategies for dynamic resource allocation algorithms in multibeam satellite systems. His research interests include THz communications, satellite communications, satellite constellations, and system design for communications and sensing.*

Juan Jose Garau-Luis *is a Ph.D. Candidate in the Department of Aeronautics and Astronautics at MIT. In 2017, under the CFIS program, he received two B.Sc. degrees in Telecommunications Engineering and Industrial Engineering from Universitat Politecnica de Catalunya, in Spain. In 2020, he received his M.Sc. degree in Aerospace Engineering from MIT. He has worked as a data scientist and researcher at Google Brain, Novartis, Arcvi, and the Barcelona Supercomputing Center. His research interests include autonomous systems, Machine Learning, and communication networks.*

Prof. Edward Crawley *received an Sc.D. in Aerospace Structures from MIT in 1981. His early research interests centered on structural dynamics, aeroelasticity, and the development of actively controlled and intelligent structures. Recently, Dr. Crawley's research has focused on the domain of the architecture and design of complex systems. From 1996 to 2003 he served as the Department Head of Aeronautics and Astronautics at MIT, leading the strategic realignment of the department. Dr. Crawley is a Fellow of the AIAA and the Royal Aeronautical Society (UK), and is a member of three national academies of engineering. He is the author of numerous journal publications in the AIAA Journal, the ASME Journal, the Journal of Composite Materials, and Acta Astronautica. He received the NASA Public Service Medal. Recently, Prof Crawley was one of the ten members of the presidential committee led by Norman Augustine to study the future of human spaceflight in the US.*

Dr. Bruce Cameron *is a Lecturer in Engineering Systems at MIT and a consultant on platform strategies. At MIT, Dr. Cameron ran the MIT Commonality study, a 16 firm investigation of platforming returns. Dr. Cameron's current clients include Fortune 500 firms in high tech, aerospace, transportation, and consumer goods. Prior to MIT, Bruce worked as an engagement manager at a management consultancy and as a system engineer at MDA Space Systems, and has built hardware currently in orbit. Dr. Cameron received his undergraduate degree from the University of Toronto, and graduate degrees from MIT*