

# Learning from History: Scoring & Automating Spacecraft Constellation Schedules

Luís F. Simões<sup>1</sup>, Ben Day<sup>2</sup>, Vinutha Magal Shreenath<sup>3\*</sup>, Callum Wilson<sup>4†</sup>,  
Christopher P. Bridges<sup>5</sup>, Sylvester Kaczmarek<sup>6</sup>, Simone Fratini<sup>7</sup>, Bruno Sousa<sup>8</sup>, Alessandro Donati<sup>8</sup>

<sup>1</sup>*ML Analytics, Portugal*

<sup>2</sup>*Department of Computer Science and Technology, University of Cambridge, UK*

<sup>3</sup>*AI Sweden, Sweden*

<sup>4</sup>*Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow, UK*

<sup>5</sup>*Surrey Space Centre, Department of Electronic Engineering, University of Surrey, UK*

<sup>6</sup>*Imperial College London, UK*

<sup>7</sup>*Solenix Deutschland GmbH, Germany*

<sup>8</sup>*European Space Agency, European Space Operations Centre (ESA/ESOC), Darmstadt, Germany*

**The scheduling of spacecraft operations is a complex problem involving multiple objectives, some of them hard to capture explicitly. Spacecraft operators rely on a trove of experience to build such schedules. However, the process is time consuming, and will not scale to the needs of managing the large constellations that will soon be in operation. The Cluster-II mission, with its 4 spacecraft and over 20 years of operations, provides a wealth of data and operational knowledge that make it an ideal test case over which to develop approaches for schedule evaluation and construction. In this work, operator objectives were formalized into a number of scoring functions that can be used to obtain real-time feedback during manual or automated schedule construction. Additionally, by processing years of Cluster-II operational data, a machine learning model of operators' decision making was built, thus enabling the capture of tacit knowledge missing from the scoring functions. These approaches are currently being integrated into ESA-ESOC's planning tools and their use in real operations is imminent at the time of writing.**

## I. Introduction

Effective mission operations are crucial to realising successful space missions. *Operations* encompasses various activities to ensure the safety and continuous operational availability of a mission. One of the tasks facing mission operators is ensuring the data collected on board spacecraft are downlinked to Earth. This requires scheduling 'passes' where a communication link is formed between a spacecraft and a ground station to offload stored data. Scheduling passes is a multi-objective optimisation problem with goals such as minimising total communication time, maximising scientific data return, and ensuring the robustness of the schedule to minimise data loss.

The problem of scheduling passes applies to any mission collecting and storing data on-board. Here we consider operations of the Cluster-II constellation which is operated by the European Space Agency (ESA). The mission consists of four spacecraft which monitor features of the Earth's magnetic field and, in particular, the effect of solar winds. Each of the four spacecraft have 11 instruments on board which together collect scientific data continuously. 2019 marked the 20<sup>th</sup> year of the mission's original 2-year planned lifetime and its data have been used in over 3000 scientific publications. The scheduling problem for Cluster-II presents many interesting challenges since it involves multiple spacecraft and has been active for a long time, so a solution here will be widely applicable to less challenging contexts. Currently, operators at the European Space Operations Centre (ESOC) perform the scheduling process manually. Doing so can take up to 1.5 days/week of operators' time and so automating this process would relieve the operators of a considerable burden. Here we address the problem of autonomous scheduling for the Cluster-II mission.

---

\*Work completed while affiliated with KTH, Sweden.

†Correspondence to callum.j.wilson@strath.ac.uk

The remainder of this paper is organised as follows: in Section II we describe the background to this problem and in Section II.A previous attempts to incorporate automation. We then describe our novel approach of evaluating schedules and modelling human decision making in Section III. Section IV states the results obtained by applying our methodology and Section V presents conclusions and recommendations for future work.

## II. Background

All four spacecraft in the Cluster-II constellation collect data continuously. Their on-board data handling consists of a *circular buffer*, a kind of looped storage system where the oldest data is overwritten by the latest. The overwritten data cannot then be retrieved. Therefore, the main objective of scheduling is to prevent data loss by regularly ‘downlinking’ data from the spacecraft. During a downlink operation, a spacecraft creates a communication link with a ground station on Earth which receives its data and ensures that the data being overwritten is not lost. We refer to the amount of data stored on-board a spacecraft that has not been downlinked as its *fill level*. There are physical constraints that determine whether a spacecraft can communicate with a given ground station; most simply, for example, the spacecraft must not be below the horizon of the ground station, nor at so large a distance to prevent effective communication with the power available on board. A time period where a spacecraft can communicate with a specific ground station is called a *visibility* and where a downlink is booked (always within a visibility) is a *pass*. A full *schedule* is then a series of passes for each spacecraft, which are typically made week-by-week up to 4 weeks in advance.

With this simple description of the problem, it seems trivial to create schedules which downlink whenever possible. However, there are many other constraints and objectives associated with the mission. The *link budget* is dependent on various physical factors to do with the spacecraft’s orbit and the ground station parameters. This budget dictates the rate at which a spacecraft can downlink data – either in Low bit rate (LBR) or High bit rate (HBR). To minimise communication time, it is desirable to always communicate in HBR mode. Every visibility will have a different link budget trend which affects the overall schedule.

As noted previously, Cluster-II is now quite an old mission which means it does not have high priority in booking passes. This means its visibility windows are often unavailable for certain times due to other missions’ bookings. The mission operation’s budget has become tighter over the years as well, so the operators must try to minimize the monetary cost of a schedule. This cost is affected by the total duration of passes and how well the passes align with operator’s shifts, which dictates the amount of overtime required to fulfil a schedule.

So far we have discussed general constraints and objectives which are mostly common to any space mission. Cluster-II also has several unique challenges associated with its scheduling. Prolonged exposure to radiation has caused all the spacecrafts’ batteries to stop working and so they rely directly on their solar panels. During an eclipse, this means they must shut down entirely – clearing their memory and not collecting any data. Any data left on the spacecraft before an eclipse is lost, so the operators attempt to ensure all data is downlinked before an eclipse. It is usually not possible for multiple spacecraft to communicate with one ground station at the same time, which must be considered when making a schedule. However, if the angular separation between two of the Cluster-II spacecraft is sufficiently small, they can simultaneously downlink data to a ground station. This is referred to as a Multiple Spacecraft Per-Aperture (MSPA) pass. The operators try to maximise the number of MSPA passes since they are far more cost efficient – effectively two passes for the price of one.

### A. TIAGO

It currently takes operators between a few hours and 1.5 days to create these schedules per week, depending on their experience with the task and the particular difficulty of scheduling that week. To make the pass planning process more efficient, to try to reduce human workload and to avoid any loss of scientific or housekeeping data a tool based on AI technologies, TIAGO, Tool for Intelligent Allocation of Ground Operations, has been developed as a collaboration between the Cluster-II Flight Control Team and the Advanced Mission Concepts section at the European Space Agency [1]. The approach chosen was to model the problem as a planning problem and to use a planner deployed on top of the APSI (Advanced Planning and Scheduling Initiative) platform [2]. The solving technology is based on timeline planning, an approach well consolidated to provide advanced solutions to support space operations [3]. One of the reasons for this approach is the capability of enabling, in a flexible way, the integration of planning and scheduling. Moreover, various software development environments exist for rapid prototyping, test and synthesis of new planning and scheduling applications based on timeline planning. An integration of TIAGO with the mission ground segment software has been implemented to automatize the generation of the planning problem from the mission data and the translation of the plan back to the mission database.

TIAGO is essentially a constraint-based temporal planning and scheduling system designed to allow the right mix between exploration of the search space (i.e., diversification of the solutions generated) and its exploitation (i.e., optimization of the plans). In particular two aspects were considered as optimization criteria: the fill level of the on-board Solid State Recorder and the ratio between tracking hours and booked hours. The first criterion measures the robustness of the solution (fill level too close to the memory capacity can be a problem in case of missing passes) whereas the second one gives an estimation of the cost-efficiency of the produced plans.

An operational evaluation [4] had shown that besides the vast amount of mostly static constraints that have to be strictly satisfied for each planning period (for instance on-board memory production and consumption, ground station acquisition/release/switch logic and varying download bitrates profiles), there exist various soft constraints whose satisfaction is often very flexible and a decision has to be considered in a case by case evaluation. Given that, the pass planning problem for Cluster-II, is a multi objective optimization problem, with conflicting optimization criteria (plan cost and robustness). Various operational aspects, for their complexity, are not captured by the optimization process. Issues are identified by the operator on a weekly basis, for instance more robustness is requested when there is low visibility, or reduced costs are preferred on the basis of specific local requests from ground station managers. Also the complexity of the problem is not general and depends on the week considered. Last but not least, it is the operator that ultimately decides to take more or less risky plans on the basis of various contingencies that cannot be stated in the domain description or optimization criteria.

As a consequence, although able to produce valid pass plans, optimized with respect to classical planning and scheduling criteria, like cost and robustness, TIAGO did not meet expectations in terms of other criteria, like choice of ground stations, duration of allocated passes, “smoothness” of pass allocation, “flatness” of memory allocation, and others. All these features were either not represented in the model or even not considered when the model was designed.

This demonstrates an additional challenge in automating scheduling: the operators cannot specify all the constraints necessary to produce a satisfactory schedule despite being able to judge whether a schedule is satisfactory. Therefore, some of the operators’ scheduling knowledge of how to meet their objectives must be *tacit* [5] \*, meaning it cannot be easily expressed or transferred. To cope with these issues, a different approach has been tried. Instead of trying to explicitly represent in the model constraints objectively difficult to define and enforce, given the availability of years of plans generated for Cluster-II, we have tried to learn these features.

### III. Methodology

We based our methods on the historical schedule data provided by ESA operators and demonstrations of the current scheduling process. In the historical data, there are files with time-series data for visibility windows available, passes booked by other spacecraft, passes which were booked for cluster spacecraft, each spacecraft’s science modes and fill level, opportunities for MSPA, and eclipses. This allowed us to analyse the operator-made schedules for various situations and try to determine any patterns. The issue with this data analysis is it lacks information on the individual decisions made in creating the schedules. In the demonstrations given by operators, we saw their decision making process while scheduling. This gave the opportunity to extract their concrete objectives and gain insight into their more abstract goals, or tacit knowledge [6].

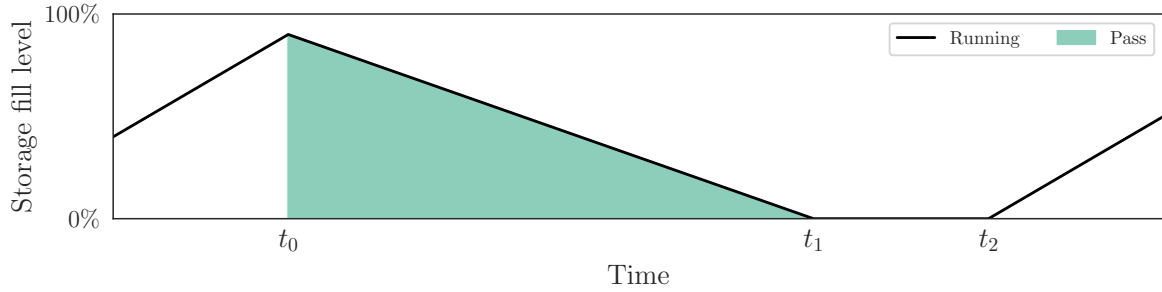
Our first goal was to create a way of objectively evaluating schedules. This could not simply be the satisfaction of hard constraints, but must describe what the operators intend to optimise. Using the measurable parameters given in the data, such as fill level and pass duration, we formalised metrics which assess how ‘good’ a schedule is in terms of these parameters. We refer to the mathematical expressions of these metrics as ‘scoring functions’. Ideally, a schedule which is optimal for these scoring functions is satisfactory to an operator.

The scores attempt to capture the operator decision making process without considering individual decisions. Alongside these scores, we also sought to model operator decision making directly. This model would take as input the information seen by operators and output some decision, such as whether or not to book a pass. A sufficiently accurate model could be used to replicate previous schedules and produce new ones which achieve the same standard as those created by operators.

Both the scoring functions and human model form a basis for automated scheduling. In the following sections we describe the methods used to build these modules and how they were brought together to create a scheduler.

---

\*As opposed to the explicit knowledge they are able to share easily.



**Fig. 1 Inefficient pass booking example.** In this example a pass is booked between  $t_0$  and  $t_2$ . From  $t_1$  the storage is empty and the download speed is limited to the rate of data collection. This is reflected in a decrease in the cost efficiency score.

### A. Scoring Functions

These scoring functions formalise several aspects of the constraints and objectives and attempt to make the operators' aims explicit. We list them here with a brief description of the objective they each represent.

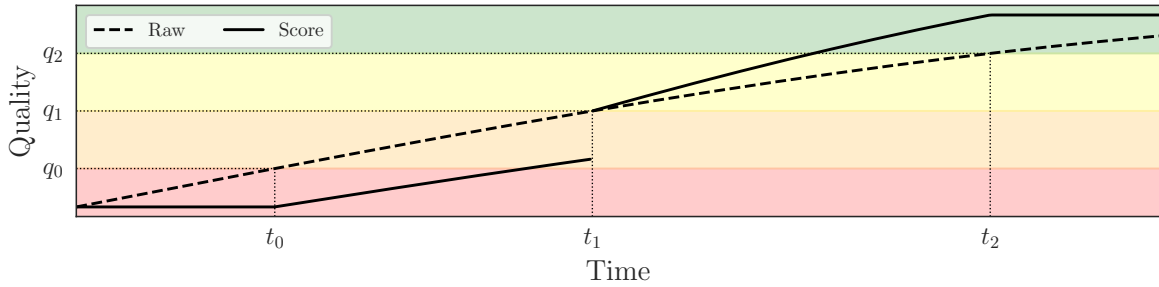
- *Cost Efficiency*  
The schedule should be as cost efficient (amount downloaded / cost) as possible.
- *Loss (inefficient data collection)*  
A pass should be no longer than necessary to downlink all of the spacecraft's data.
- *Link Budget Alignment*  
The link budget should always be in HBR and maintain a margin for error above the HBR threshold.
- *Fill Level*  
The fill level should be as low as possible and stay below a threshold value.
- *Fragmentation*  
Scheduled passes should not be spread out over long periods of time but should be clumped together.
- *Shift Work*  
Scheduled passes should align with shift times of operators.
- *Uplink Consistency*  
Uplink passes should occur regularly.
- *MSPA*  
MSPA should be used whenever possible.
- *Eclipse Loss*  
Data loss due to eclipses should be minimised and uplink should occur as close as possible to the start and end of an eclipse.

Each function takes as input some measurable aspect of a schedule and returns a 'score' - a scalar value between 0 and 1. This score indicates how well the schedule fulfils the respective function's objective with 1 and 0 being the best and worst possible scores respectively. Some of the functions apply to individual passes whereas others apply over a given time-range either for all or one of the spacecraft. We want to evaluate all the scores over a fixed time-range for all spacecraft which for certain scores requires averaging (or otherwise statistically evaluating) scores over multiple passes or spacecraft. The expressions used to do this are given in Table 1, and illustrated in Figures 1 - 8

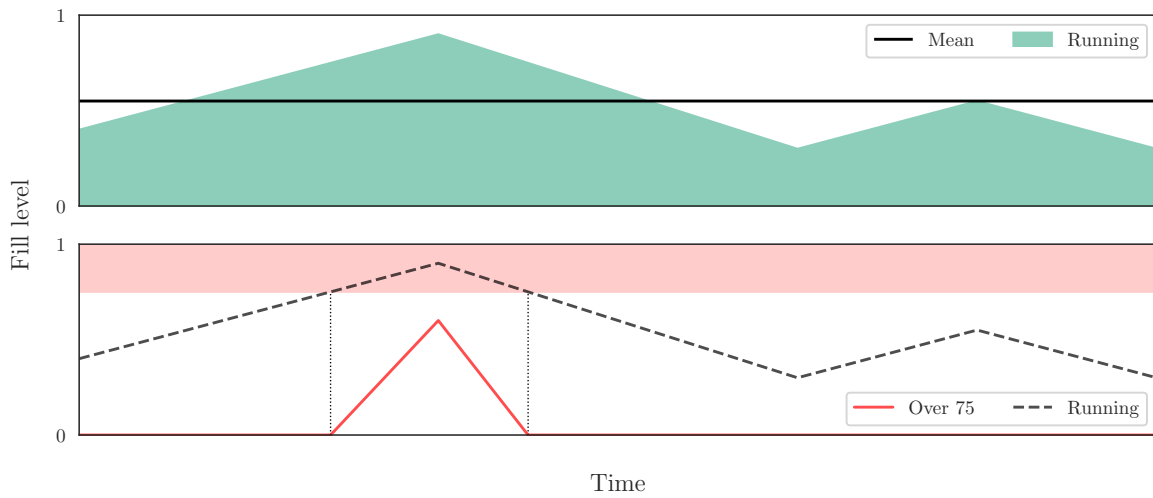
### B. Human Model

To develop a model which takes actions as a human would, we must first define the inputs to this model. These inputs should represent what the human operator sees when making a schedule. Here we refer to this input as a 'context' in which a decision is made. When assessing whether to book a pass for a given spacecraft within a specific visibility, we assumed the most important parameters affecting an operator's decision to be as follows:

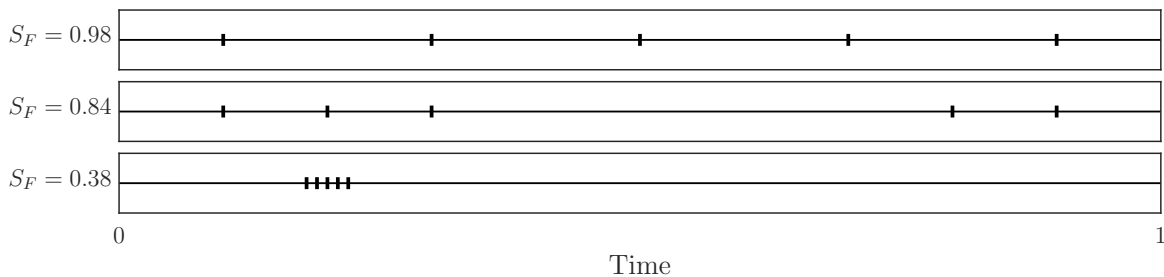
- **Fill level** of the spacecraft memory at the beginning of the visibility;
- **Link budget** during the visibility;
- **Duration** of the visibility.



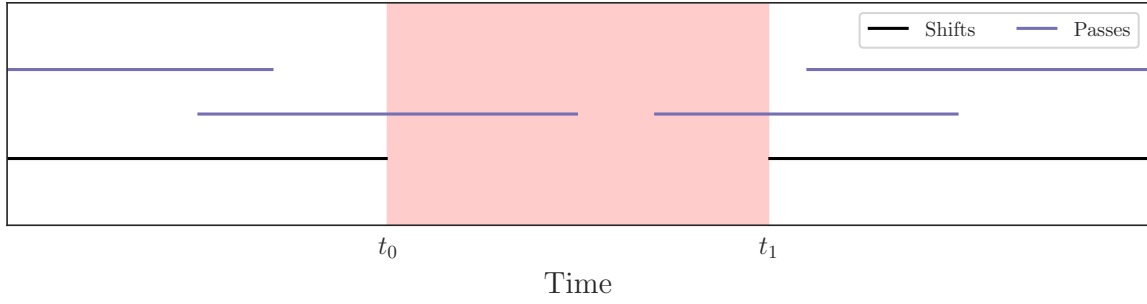
**Fig. 2** Scoring the quality of the link budget during a visibility. Below some threshold  $q_0$ ,  $t < t_0$ , it is not possible to connect to the spacecraft, and the score is 0 (red band). At an intermediate quality,  $q_0 < q < q_1$ ,  $t_0 < t < t_1$ , a lower-quality connection is possible (orange band). Above a further threshold,  $q_1$ ,  $t_1 < t < t_2$ , a higher-quality connection is possible (yellow band). Beyond a final threshold,  $q_2$ ,  $t_2 < t$ , further increases in quality do not register in the link budget, as seen in the score reaching 1.



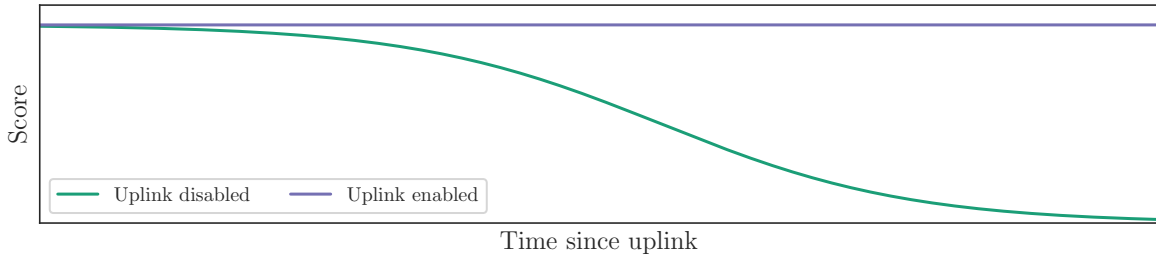
**Fig. 3** Components of the fill level scoring. It is desirable to maintain a low mean fill level, shown in the upper panel. It is especially important to not allow the storage to reach high fill levels, and a heuristic used by operators is to treat 75% full as the acceptable upper limit. This can be scored by additionally penalising allowing the fill level to exceed 75%, shown in the lower panel.



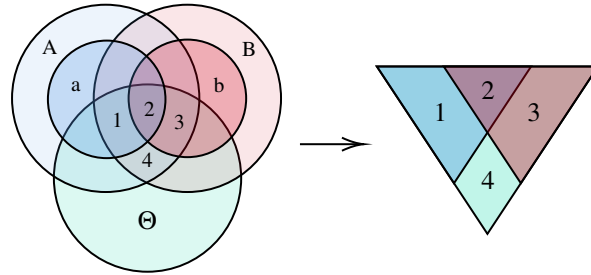
**Fig. 4** Fragmentation scores for three examples of events. The top panel shows evenly spaced events with a high fragmentation score,  $S_F = 0.98$ . The middle panel has events in two equally spaced clusters, for an intermediate score of 0.84. The bottom panel has events tightly clustered towards the start of the interval, achieving a low score of 0.38.



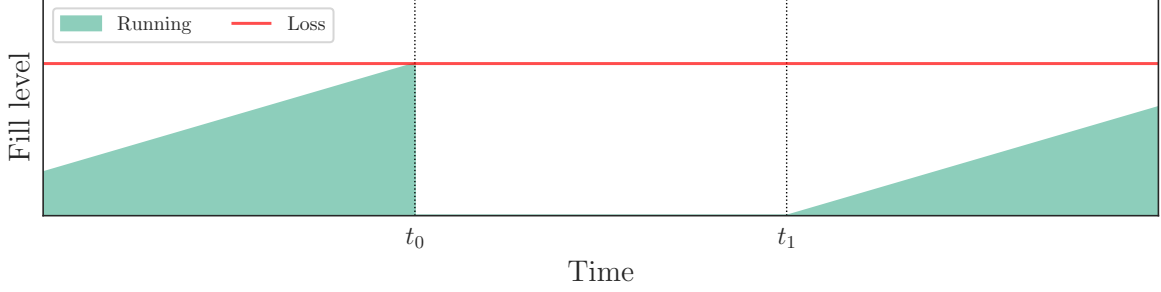
**Fig. 5** Passes and operator shifts. Scheduling passes to overlap with operator shifts is preferable, passes scheduled outside of shift hours (red) disrupts operators.



**Fig. 6** The relative scores of stations with and without uplink capabilities following an uplink enabled pass. Uplink connections provide instructions to the spacecraft. Immediately after an uplink enabled pass there is no need to provide new instructions, and this is reflected in the scores being equal. As time goes on it becomes increasingly important to upload new instructions, and the score for passes without the option to uplink drops.



**Fig. 7** Set diagram showing the possible combinations of MSPA, visibilities and passes for two spacecraft,  $a$  and  $b$ . The bottom circle represents the set of times when MSPA is possible (angular separation  $< 0.03$ ). The blue sets labelled  $A$  and  $a$  are the times of visibilities and passes for spacecraft  $a$ , with all passes occurring during visibilities, and similarly for the red sets for spacecraft  $b$ . The part of the plot we are interested in is the central section, with labels (1, 2, 3, 4), enlarged in the triangular representation (right). Possible MSPA times are given by the intersection  $A \cap B \cap \Theta = 1 \cup 2 \cup 3 \cup 4$ . Created MSPA times are given by  $1 \cup 2 \cup 3 = (A \cap b \cap \Theta) \cup (a \cap B \cap \Theta)$  i.e. those times when a pass was booked for spacecraft  $a$  and spacecraft  $b$  is visible during MSPA, and vice versa. Used times are given simply by  $2 = a \cup b \cup \Theta$  – passes are booked for both craft at the same time during MSPA. The scores are then the ratios of these: created / possible and used / created. There is no penalty when MSPA is not possible.



**Fig. 8** During an eclipse,  $t_0 < t < t_1$ , the satellite powers off and any stored data is lost. The penalty is proportional to the lost data.

**Table 1** Mathematical formulations of the scoring functions.  $\Delta t$  refers to the length of the interval being scored (a whole schedule, a month etc.),  $\Delta t_p$  refers to the length of a pass being scored,  $\Delta t_{prev}$  is the time since the previous pass,  $t_0$  is a parameter. For those scores that apply to individual passes, an aggregate score can be formed, either with a simple average or using quantiles as appropriate (for example, we may want our scores to be more strongly influenced by weak outliers to better highlight deficiencies). The *events* over which the fragmentation sum is taken refers to any operation made, including the start and end of passes and the start and end of the considered time period. The measure is based on entropic arguments, adapted from work on measuring clumping [7]. Booking MSPA is a two stage procedure – booking the first (*created*) and second (*used*) passes. The eclipse loss is, in principle, only the data lost at the time the eclipse starts, hence the first score, but we can *smooth* this step function with leading and trailing sigmoidal curves to better represent the impending loss.

Score	Function	Related figures
Cost Efficiency	$\frac{\text{download total}}{\text{cost}(\Delta t_p)} \times \frac{\text{cost}(1\text{hour})}{\text{fast download rate} \times \Delta t_p}$	1
Link Budget Alignment	$\text{mean}_{\Delta t_p}$ (normalised link budget)	2
Fill Level	$1 - \text{mean}_{\Delta t}$ (fill level)	3
Fragmentation	$-\sum_{\text{events}} \frac{t_i - t_{i-1}}{\Delta t} \log_{(n+1)} \left( \frac{t_i - t_{i-1}}{\Delta t} \right)$	4
Shift Work	$\frac{\text{pass time scheduled during shifts}}{\text{total scheduled time}}$	5
Uplink Consistency	$1 - \frac{e^{r \cdot \Delta t_{prev}}}{e^{r \cdot \Delta t_0} + e^{r \cdot \Delta t_{prev}}}$	6
MSPA	$t_{\text{created}} / t_{\text{possible}}$ $t_{\text{used}} / t_{\text{created}}$	7
Eclipse Loss	$1 - (\text{fill level at eclipse})$ $\frac{\text{fill}}{1 + e^{-k(\Delta t_b + c)}} ; \frac{\text{fill}}{1 + e^{k(\Delta t_a - c)}}$	8 -

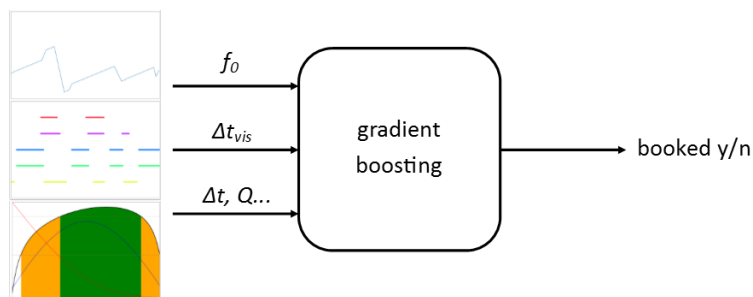
By expressing these parameters numerically in a ‘context vector’, we can use this as an input for our model to decide whether to book a pass in a given visibility. The initial spacecraft fill level and visibility duration are scalar values which can easily be incorporated to the context vector, however the link budget is time-series data which must be processed. We take the following statistics from the link budget to include in the context vector: duration of LBR, duration of HBR,

quartiles, and maximum. The context vector,  $x$  for making a decision on a given visibility is as shown:

$$x = \{f_0, \Delta t_{vis}, \Delta t_{LBR}, \Delta t_{HBR}, Q_1(P_{RX}), Q_2(P_{RX}), Q_3(P_{RX}), \max(P_{RX})\}$$

Where  $f_0$  is the spacecraft’s fill level at the start of the visibility;  $\Delta t$  denotes a time range with the subscripts  $vis$ ,  $LBR$ , and  $HBR$  indicating duration of the visibility, time for LBR transfer, and time for HBR transfer respectively; and  $P_{RX}$  is the link budget over the visibility as described by the statistics noted previously. This choice of context vector describes an individual visibility, however when operators make schedules they can view multiple visibilities to decide in which to book a pass. This would suggest that to accurately model the operator’s decision making, context from neighbouring visibilities should be incorporated. However, based on our analysis presented in Section IV, this single visibility horizon view is sufficient to model the operator’s behaviour.

A machine learning model, specifically a Gradient Boosting Machine, was trained over the available historical data. It was tasked with learning a mapping from  $x$  representations of past visibilities to the decision of whether or not human operators booked passes within those visibilities. Figure 9 shows a black-box representation of the human model with its inputs and single decision output.



**Fig. 9 Black-box representation of the human model showing inputs (what the human operator sees) and outputs (the potential actions for the human operator).**

### C. Autonomous Scheduler

Using scoring functions and a model of operator decision making, we can create schedulers which either optimise a combination of the scoring functions, or make decisions similar to a human operator. The simplest approach to optimising a schedule is to do a grid search and find the solution with the best score. Here the problem space is too vast for such an approach due to the number of visibilities within a week for each spacecraft (up to 200) and the problem of pass placement within a visibility. A more efficient approach, which has been applied to other scheduling problems, is beam search [8, 9]. This algorithm is similar to a breadth-first tree search, where at each level the search moves down to only a finite number of promising nodes. In our problem, this evaluation of nodes can be implemented through the scoring functions, the human model, or both.

## IV. Results

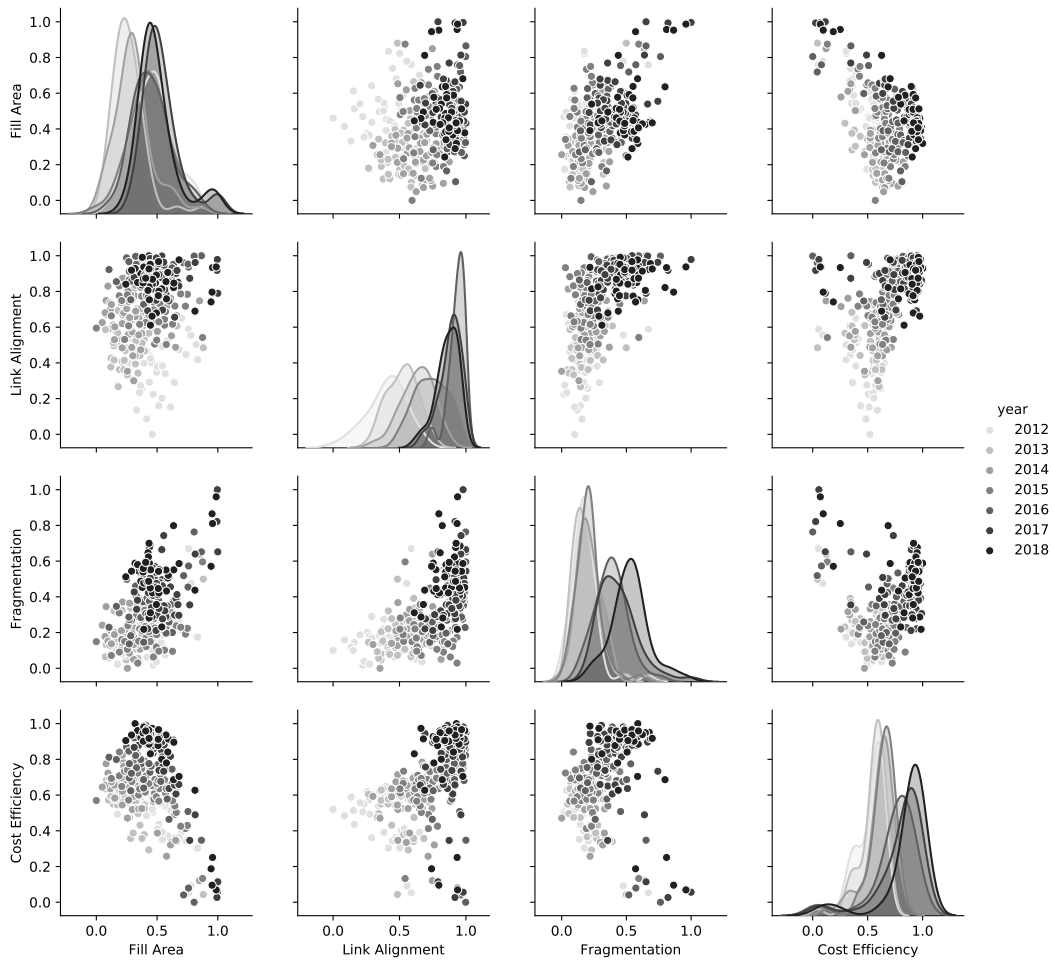
Here we present the results from the analyses we conducted on historical data and the application of our autonomous scheduler to the problem. Along with the results presented here, this project produced several outputs which are of use to the operations team at ESOC. We also describe these outputs in this section.

### A. Historical Data Analysis

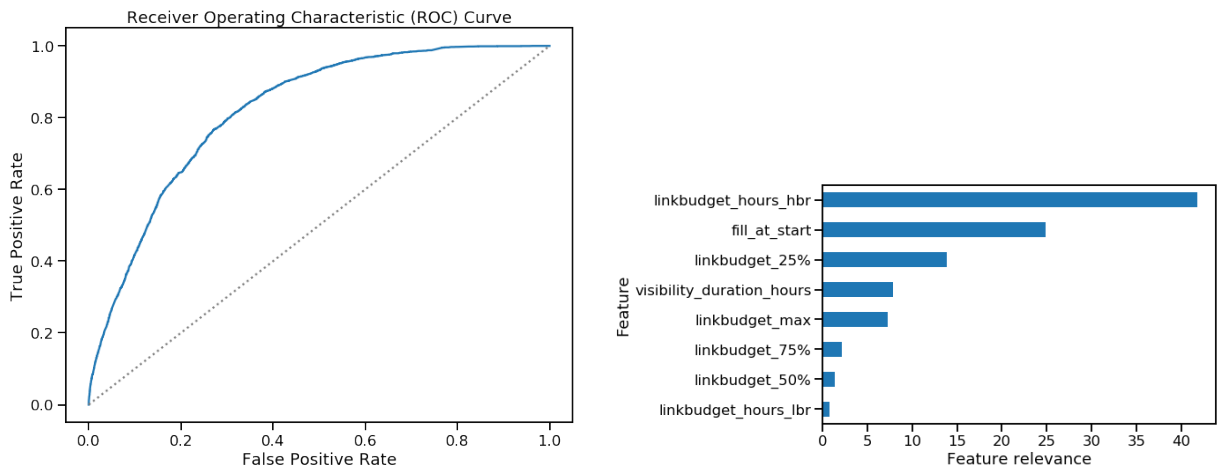
With the scoring functions developed, we applied these to the historical operator-generated schedules to observe any emergent trends with respect to these scores. For this analysis, scores were taken over one week – the typical horizon over which operators create schedules – from 2012 to 2018, which gives a distribution in the scores for each week over a year.

We compare the individual scores and the relationships between them over the given time range with a pair plot shown in Figure 10. For simplicity, here we consider 4 of the scoring functions: fill area, link alignment, fragmentation, and cost efficiency. The axes on the diagonal show distributions of individual scores for each year and other axes in the

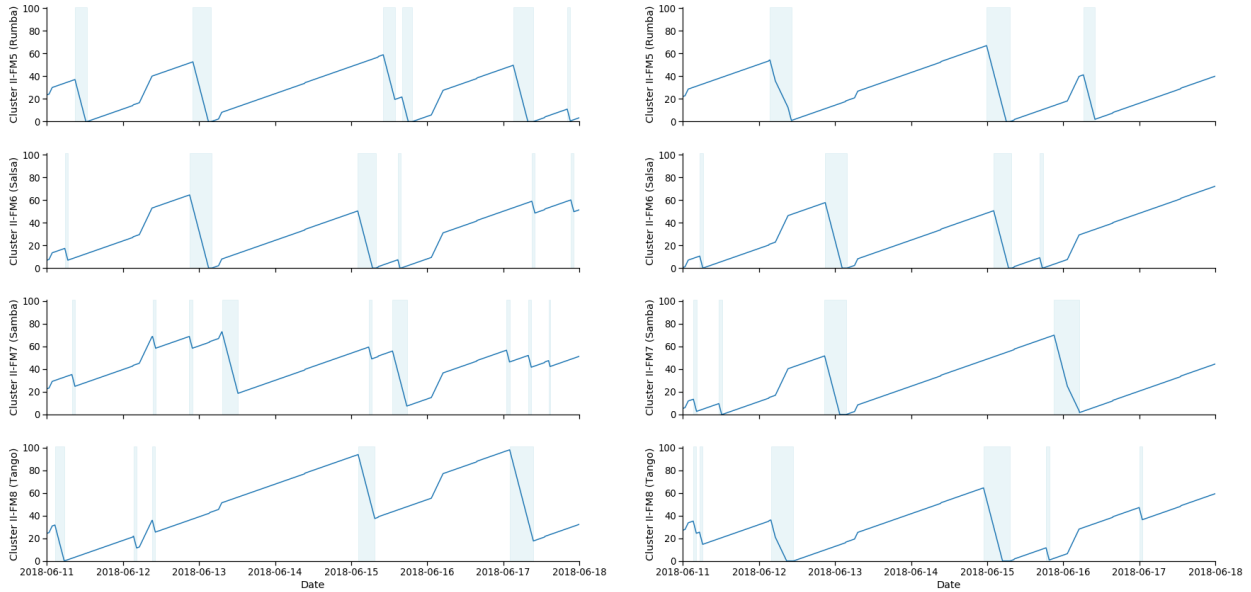




**Fig. 10** Pair plot showing variations of four scores and their relations. Shade indicates the year over which scores are taken and each point represents a score for one week.



**Fig. 11** Feature relevance and ROC curve for the gradient boosting binary classifier.



**Fig. 12 Comparison of a real schedule booked by operators (left) and a schedule for the same week created autonomously using a beam search (right).**

figure show scatter plots of two scores. In both cases, the shading indicates the year over which those scores occurred. To normalise the scores and ensure they all occupy the same range, since some scores tend to be better in general than others, we take the Z-score over the weeks. From this analysis, we make two key observations:

- 1) The scores tend to improve over time;
- 2) Most of the scores are not correlated with each other.

There are several reasons the scores show an increase in later years, such as more favourable orbits and changes in the mission requirements which align with our scoring functions, but might not have been present in the past. Additionally, the factor of human improvement over time through experience and practice could certainly play a role. Since the scatter plots show a lack of correlation between scores and a spread for any given year, it is also clear that there are spaces of solutions to explore rather than unique, optimal solutions. For example, when optimising a schedule for these scores there could be a trade-off between fill area and cost efficiency, where different valid solutions yield different scores for both of these aspects.

Next we validate our human model on the historical schedules. Figure 11 shows the Receiver Operating Characteristic (ROC) curve for the gradient boosting model, which has a train set ROC AUC score of 0.82 (0.79 ROC AUC estimated through 10-fold cross validation). Despite the highly simplified context being given to the model, which is not a complete description of what the operator sees, this classifier can distinguish between visibilities which were and were not booked to a useful degree of accuracy. By looking at the feature relevance chart also we see which features are most significant when making a classification. The model mostly considers the amount of HBR time in a pass and the next most important feature is the fill level, which is consistent with our observation of operators' decision-making.

## B. Autonomous Scheduling

The historical data analysis suggests our approach of of scoring functions and a human model can lead to schedules of a similar standard as those of human operators. Unlike the quantitative analysis of existing schedules, we can only evaluate our new schedules qualitatively by comparing them to those made by operators. As discussed previously, the state-space of possible schedules includes a multitude of solutions that evaluate to equally good trade-offs in objective space. Therefore it may not be necessary to exactly replicate the previous schedules created by operators, however at this stage such a comparison gives useful feedback on the approach.

Figure 12 shows the schedule booked by operators for a specific week and a schedule created using the beam search method for the same week. Very broadly speaking, there is little to distinguish between the human-made and automatically generated schedules. In particular, for this week the beam search solution for spacecraft 2 looks very

similar to that which the operators used. This suggests our approach has merit and motivates a more objective analysis of the automatic schedules.

### C. Project Output

Here we have described the results from our analyses, which only represents part of our contribution in this project. We set out with the aim of not only creating an autonomous scheduler, but to create useful tools for the Cluster-II operators at ESOC. Now we will show how we intend for our work to be integrated into the current mission operations.

**Instant feedback on operator schedules** Currently, when operators create schedules, they effectively do so ‘blindly’ with limited assistance or feedback from the scheduling interface. Using our scoring functions, we can display how well a schedule performs to an operator while they make the schedule. This is beneficial both for guiding current operators’ decisions and for training new operators who lack the same level of experience scheduling for this mission. Work has already taken place along these lines, and is reported in the next section. The human model can also be used to indicate which visibilities an operator is likely to select. This would accelerate the scheduling process substantially by saving operators time otherwise spent navigating various parts of the application interface to select visibilities.

**Tools for schedule simulation** One of our major tasks in this project was to implement a scheduling environment that accounts for the many complexities of the Cluster-II mission. With a simulator and automated scheduler in hand, mission operators can play out “what-if” scenarios well into the future, and obtain feedback from how operational decisions will impact things such as amount of scientific data collected over time.

**Scalable autonomous scheduling** Our approach to this scheduling problem takes into account the many idiosyncrasies of Cluster-II, but these only apply to this mission. In the future, there will be many larger constellations of satellites requiring downlinks and the current system of manual scheduling does not scale to this. Here we have investigated an autonomous scheduling approach that can be adapted to varying mission requirements and scaled up to larger constellations of satellites.

### D. Integration with OpsWeb, ESA-ESOC’s Planning Tool

Prior to integration with the planning tools at ESOC the proposed scoring functions developed in the context of this study were reviewed and re-iterated with the Operations team. The goal was to fine-tune them to have more relevance and meaning for the planning activity. The final set of scoring selected included:

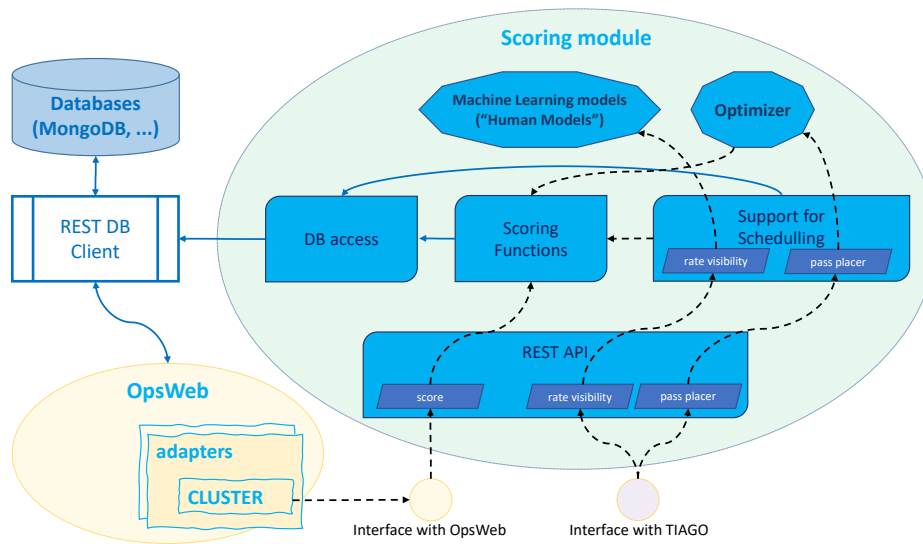
- 1) *Shift Fragmentation* – scores higher when passes are bundled together to avoid fragmentation of operators shifts.
- 2) *Mass Memory Threshold* – Spacecraft memory should at all times be  $> 0\%$  and  $< 80\%$ . The longer, and further, spacecraft surpass these thresholds, the lower the score will be.
- 3) *Tracking efficiency* – scores highest when all data is retrieved in High Bit Rate.
- 4) *S/N Margin* – there are periods during passes where the Signal to Noise level received at the station is marginal (below 0.5 dB w.r.t. threshold). If this happens for more than 20 minutes then this affects the score negatively.
- 5) *TC Uplink frequency* (new) – scores high when there are at least 2 commanding passes per orbit (to be sure it is possible to uplink all the time-tagged commanding).
- 6) *Ranging Needs* (new) – scores high when the Flight Dynamics ranging requirements are fulfilled.
- 7) *MSPA Usage* – scores high when usage of MSPA (Multiple Spacecraft per Aperture) opportunities is maximized.

The scoring period was fixed to be equivalent to a short term planning period. These are defined well in advance with the Science Operations team and include 3 to 4 orbits of 54 hrs each — so approximately one week.

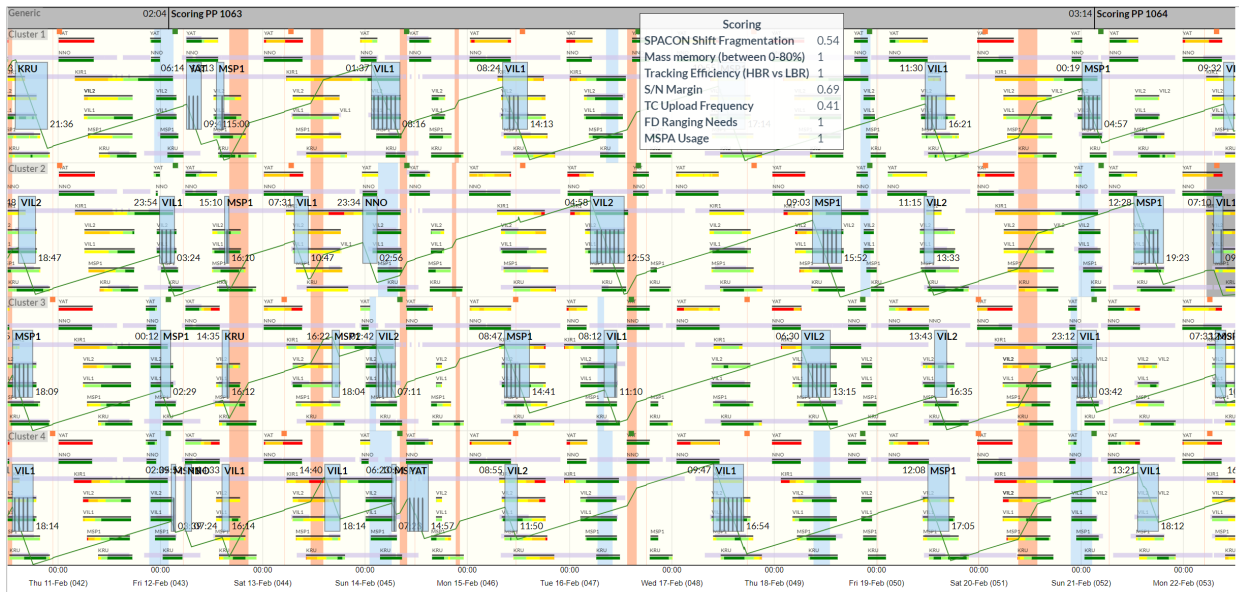
The scoring functions [10] were deployed as a standalone application interfacing the planning tool (called OpsWeb), with a REST DB client<sup>†</sup> providing access to all the planning data required to compute the scores. The results are passed directly, via a REST API, to the front-end that displays them accordingly – Figure 13.

Figure 14 shows a week’s contact plan for the 4 Cluster-II spacecraft. The horizontal bars in the background represent the periods of visibility to different stations and the red/yellow/orange/green bars show the predicted signal strength, which in turn drives the possibility to perform Ranging (the vertical bars inside contacts). The saw line represents the predicted fill level of the Mass memory. Whenever a new contact is added the Scoring is updated in real time giving timely feedback to the planner, thus facilitating the optimization of the plan.

<sup>†</sup><https://gitlab.com/librecube/lib/python-rest-db-client>



**Fig. 13 Architecture of the final application and integration with ESA-ESOC tools.**



**Fig. 14 Example of a week's planning for the Cluster-II mission, in ESA-ESOC's OpsWeb tool. As the operator books communication links the scoring functions provide real-time feedback with respect to multiple objectives.**

Testing is still on-going in particular to gain confidence on the meaning of the scores, how they are affected by changes in the plan, and how they affect the trade-offs that a planner does among the different criteria for optimization. The results so far are very promising and a full usage in real operations is imminent at the time of writing.

## V. Conclusions

Here we have developed a framework for autonomous scheduling which can be applied to the Cluster-II scheduling problem. Using lessons learned from previous attempts to automate this task, we devised two approaches. First, formalising an operator's objectives in scoring functions which can be optimised. Second, extracting information about the operator's decision making process to create a machine learning model. This approach shows merit for creating schedules to the same standard as a human would. The outputs from our project will be useful to operators at ESOC to speed up their scheduling process. This project also presents several avenues for future work. Most notably, the methods we detail here can be applied to other space missions or general scheduling tasks.

## VI. Acknowledgements

This work has been enabled by Frontier Development Lab (FDL) Europe (<https://fdleurope.org>) a public / private partnership between the European Space Agency (ESA), Trillium Technologies and the University of Oxford, and supported by Google Cloud, Intel AI, NVIDIA, Airbus and Catapult Satellite Applications. The authors gratefully acknowledge the support from ESA-ESOC, in particular from the Cluster-II Flight Control Team and the Advanced Mission Concepts and Technologies Office, as well as James Parr, Jodie Hughes, Leo Silverberg and Sarah McGeehan from Trillium Technologies and other organisations and mentors who supported FDL Europe 2019. The authors also thank ESA Academy for their offer of support to present this work.

## References

- [1] Faerber, N., Fratini, S., Policella, N., Bartesaghi, M., Costa, T., De Sousa, B. T., and Donati, A., "Cluster-II: Using artificial intelligence for automated ground station scheduling," *SpaceOps 2016 Conference*, American Institute of Aeronautics and Astronautics (AIAA), 2016, p. 2595. <https://doi.org/10.2514/6.2016-2595>.
- [2] APSI, "APSI Software Distribution Web Site," , 2017. URL <https://essr.esa.int/project/apsi-advanced-planning-and-scheduling-initiative>.
- [3] Chien, S., Johnston, M., Frank, J., Giuliano, M., Kavelaars, A., Lenzen, C., and Policella, N., "A Generalized Timeline Representation, Services, and Interface for Automating Space Mission Operations," *Proceedings of the 12th International Conference on Space Operations, SpaceOps*, AIAA, 2012. <https://doi.org/10.2514/6.2012-1275459>.
- [4] Fratini, S., Faerber, N., Policella, N., Neves Madeira, R. N., De Sousa, B. T., and Donati, A., "Ground Station Pass Allocation in Operations using AI," *SpaceOps 2018 Conference*, American Institute of Aeronautics and Astronautics (AIAA), 2018. <https://doi.org/10.2514/6.2018-2726>.
- [5] Magal Shreenath, V., "Creating Knowledge with Data Science for Design in Systems," Ph.D. thesis, KTH Royal Institute of Technology, 2019.
- [6] Simões, L. F., Day, B., Magal Shreenath, V., Wilson, C., Bridges, C., Kaczmarek, S., and Gal, Y., "FDL: Mission Support Challenge," *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019. URL <http://ciml.chalearn.org/ciml2019/accepted/Day.pdf>.
- [7] Zhang, Y., Bradlow, E. T., and Small, D. S., "New measures of clumpiness for incidence data," *Journal of Applied Statistics*, Vol. 40, No. 11, 2013, pp. 2533–2548. <https://doi.org/10.1080/02664763.2013.818627>.
- [8] Sabuncuoglu, I., and Bayiz, M., "Analysis of reactive scheduling problems in a job shop environment," *European Journal of Operational Research*, Vol. 126, No. 3, 2000, pp. 567–586. [https://doi.org/10.1016/S0377-2217\(99\)00311-2](https://doi.org/10.1016/S0377-2217(99)00311-2).
- [9] Wilt, C., Thayer, J., and Ruml, W., "A comparison of greedy search algorithms," *Proceedings of the 3rd Annual Symposium on Combinatorial Search, SoCS 2010*, 2010, pp. 129–136. URL [www.aaai.org](http://www.aaai.org).
- [10] Simões, L. F., "Integration of FDL Prototype Scheduler Functions with ESOC Planning Tools," Tech. rep., 2020.