International Conference on Space Situational Awareness
Madrid, Spain

COORDINATED PLANNER ALGORITHMS

David Moreno[1], **Almudena Dominguez**[2], **Francisco Ayuga**[3], **Alfredo M. Antón**[4], **Daniel Lubián**[5], **Ignacio Grande**[6], **and Katia Caceres**[7]
[1] [2] [3] [4] *GMV, Calle Isaac Newton 11, Tres Cantos, 28670, Spain, +34918072100*
[1] *dmespinosa@gmv.com,* [2] *aldominguez@gmv.com,* [3] *fjayuga@gmv.com,*
[4]*amanton@gmv.com,*
[5] [6] *Deimos Engineering and Systems, SLU, Cl Francia Nº. 9, Puertollano, 13500, Spain,* [5] *daniel.lubian@deimos-space.com*, [6]*ignacio.grande@deimos-space.com*
[7] *Centro de Desarrollo Tecnológico Industrial (CDTI), Calle del Cid, 4, 28001 Madrid, Spain, +34915815500* [5]*katiamaria.caceres@cdti.es*

Keywords: *EU SST, Survey, tracking, optimization, machine learning*

ABSTRACT

The "Decision of the European Parliament and the Council Establishing a Space Surveillance and Tracking Support Framework" was adopted on April 16, 2014. It established the European Space Surveillance and Tracking (EU SST) Support Framework at European level, which evolved into a fully-fledged component of the European Union Space Programme adopted on 28 April 2021. EU SST contributes to the global burden sharing of ensuring the sustainable and guaranteed access to and use of space for all. Its primary objective is the provision of space-safety services, namely, to protect spacecraft from the risk of collision, to monitor uncontrolled re-entries, and to survey the in-orbit fragmentation of space objects.

CDTI, as part of the **EUSST Consortium**, is developing a Coordinated Planner (COPLA) for EU SST network of sensors. In this abstract, the description of the logic and algorithms developed in the COPLA project is presented. The objective of COPLA is to coordinate all the EU SST sensors in order to contribute and improve the SST services.

COPLA software is mainly divided in two processing chains, survey and tracking, that complements each other in order to optimize the coordination of the sensors. First of all, the **survey chain** is in charge of generating a survey strategy for optical sensors. With this survey strategy for optical sensors and the pointing for radars, visibilities against all the catalogued objects are obtained. These visibilities will be used by the accuracy gain algorithm in charge of computing the covariance improvement produced by the survey measurements.

With the computation of this accuracy gain, the **tracking chain** starts by selecting objects ordered by a given priority. This priority is obtained considering the space safety constraints at that moment, the covariance of the object (after being reduced with the survey chain), and other parameters. Then, observation opportunity slots are computed based on hard constraints and soft constraints, without a strict threshold, evaluated by machine learning algorithms. Besides, a sensor performance weighting

value will be obtained based on machine learning approach, based in the historical success of the sensor-object tuple. Then, the accuracy gain is also computed for these tracking slots with the same approach as the survey chain.

Finally, priority of the object, observation probability, sensor performance weight and accuracy gain together with sensor movement constraint are finally introduced in the cost function of the optimizer in order to select the optimal tracking slots that the complete sensor network must follow.

1 INTRODUCTION

The EU SST framework is in continue growth with new satellite owner operators, new services and new member states with their own network of sensors.

As SST network relies, in a first approach, on its SST network of sensors in order to produce all the SST products downstream, it is of extreme importance to coordinate the operations of all these new sensors in combination with the existing ones. For this reason, a Coordinated Planner tool is required.

The main goal of COPLA is the coordination of the EU SST network of sensors by the software evaluation and optimization of its operations in a routine basis. COPLA is in charge of gathering all the required information and generating the final products needed by the sensor operators.

In the following sections, a detailed explanation on the algorithms in charge of performing the evaluation and the optimization of the survey and tracking plans are described. These sections must be understood as a sequence of processes to achieve the optimized schedule of the sensor network.

2 SURVEY CHAIN

Survey chain is the starting point of the COPLA software once the different inputs from the EUSST DB and external servers such as orbits, sensor availability, Earth orientation parameters, solar and geophysical activity, object properties, etc… have been retrieved.

2.1 Survey strategy

In contrast with the radars, which has a fixed pointing and very large apertures, optical sensors have a limited Field of View (FoV) size and requires special observability constraints, such as target sun illumination or angular distance to blinding bodies. For this reason, telescopes need to make use of **survey strategies** on their survey operations. A survey strategy is a dedicated guiding law detailing the coordinates (in terms of right ascension and declination) where the telescope should point at each instant of time.

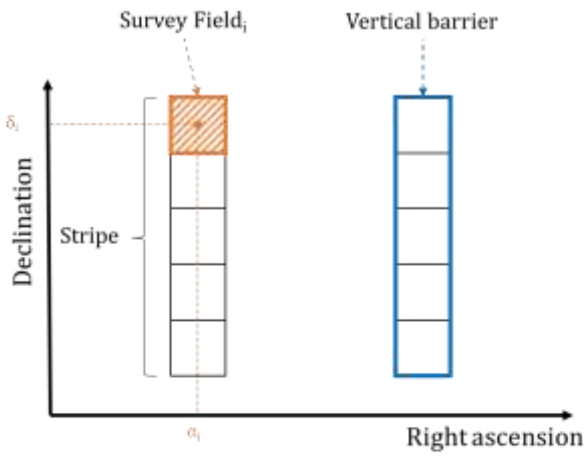

Figure 1: Survey strategy definition

COPLA software implements an optimization algorithm for selecting the most suitable strategy for each sensor based on the GEO "leak-proof" concept. That is, for a vertical stripe of desired width (in declination), the telescope time to sweep this stripe will be always shorter or equal to the time the objects require to cross the FoV assuming the typical mean motion of a GEO object $n_{GEO} = 0.25\ deg/min$.

For the optimization problem, a set of parameters derived from the sensor characteristics and the survey requirements defined need to be configured. These parameters are:

- t_{exp}: exposure time of the telescope
- t_{rep}: repositioning time of the telescope
- N_i: number of images the telescopes takes in each survey field
- N_f: number of survey fields in each stripe
- N_b: number of vertical barriers in each survey period
- γ_r: requested overlap between two stripes

The problem is then split in two smaller ones, first one related to the target declinations and second to the target right ascensions. For the declination, it has to be taken into account that the majority of the GEO objects are concentrated around $\delta = 0°$, in the so-called GEO ring, and in a non-uniform distribution around it (see Figure 2) for the non-controlled objects caused by third-body perturbations. Considering these two constraints together with the FoV and N_f, which define the declination bandwith, the algorithm will easily select the most suitable declination to place the stripe for each right ascension.

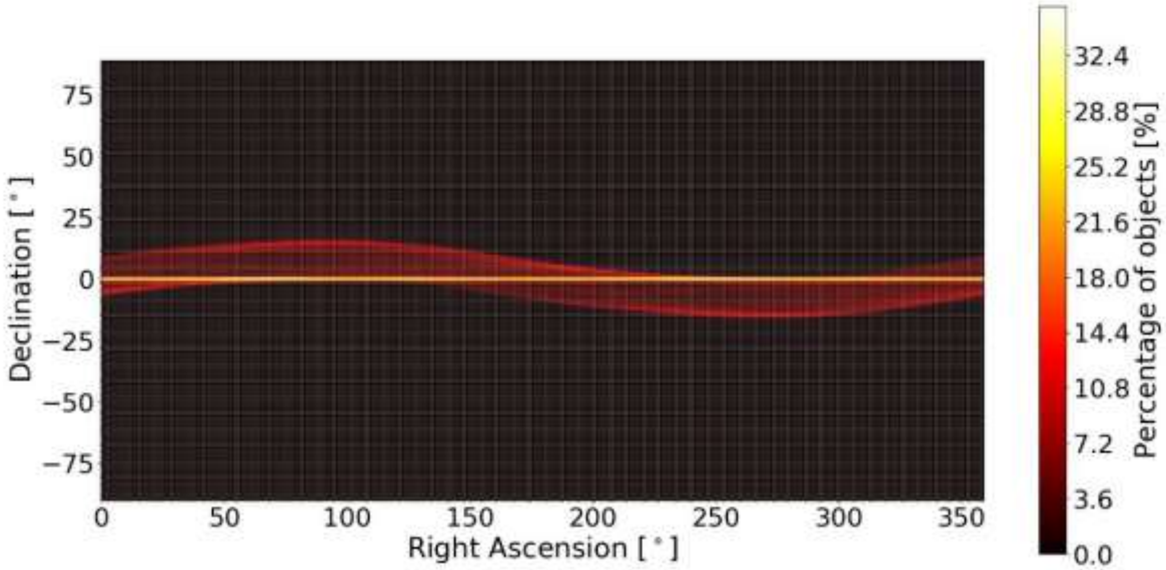

Figure 2: Density of GEO population in Right Ascension-Declination map

For the target right ascension, both the “sky constraints” and the brightness of the objects are considered. Sky constraints are known as restrictions that should be applied to the survey strategy because of saturation or poor illumination conditions or both for the telescope and the observed object. The constraints considered in this algorithm are: night period of the telescope (SZD_{min}), minimum elevation (el_{min}), maximum distance to North Galactic Pole (GDP_{max}), minimum distance to the Moon (MD_{min}) and Sun illumination restriction (umbra and penumbra). Brightness of the objects is evaluated by considering the pointing phase angle, θ (angle between Sun-Pointing Sensor-Pointing) since the properties of each object are not known at this point. After all this, the optimization problem is defined in the following way:

$$\underset{\alpha_j, \delta_j}{\text{maximize}} \sum_{i}^{N_B} \sum_{j}^{N} \frac{1}{2}\left[1 + \cos\left(\theta_{i,j}\right)\right] \Delta t_{i,j}$$

$$\text{subject to} \begin{cases} el_j > el_{min} \\ SZD_j > SZD_{min} \\ GDP_j < GDP_{max} \\ MD_j > MD_{min,NM} \text{ if } (\alpha, \delta) \in \text{New Moon} \\ MD_j > MD_{min,FM} \text{ if } (\alpha, \delta) \in \text{Full Moon} \\ \beta_j > VA_{Sun,j} + VA_{Earth,j} \end{cases} \quad (1)$$

Where:

- N_B is the number of vertical barriers and N is the total number of epochs under analysis
- $\Delta t_{i,j}$ is the survey field time for the i^{th} barrier at j^{th} epoch
- β_j is the angular difference between observations taken from the centers of the Sun and the Earth and $VA_{Body,j}$ is the view angle (half of the angular extent of the object w.r.t. certain body, also known as semi-diameter) j^{th} epoch

The physical meaning of this objective function can be understood as *brightness balanced effective survey time*. Notice that the leak-proof concept is also introduced in this optimization process by introducing the constraint of $\Delta t_i = 0$ seconds when a survey field is repeated in the same stripe, as a result of one of the restrictions is not passed.

Additionally, in the previous function it has been introduced the concept of number of vertical barriers. The physical meaning of this parameter is basically to define the number of re-observations of the observed objects. In other words, for one fixed barrier in right ascension, the sensor will sweep from East to West in local coordinates, the maximum number of observed objects will be obtained, roughly corresponding to one third of the GEO ring. On the other hand, for two fixed right ascension barriers (two sweeps), a lower number of objects will be observed with the advantage of seen these objects twice. For this last case, the cost function is subject to one additional restriction:

$$\alpha_{\mathrm{shift,i}} = t_{\mathrm{barrier,i}} n_{\mathrm{GEO}} \quad (2)$$

Where:

- $\alpha_{\mathrm{shift,i}}$ is the right ascension shift between i^{th} and $(i-1)^{th}$ barriers
- $t_{\mathrm{barrier,i}}$ is the time spent in the fixed right ascension i^{th} barrier

As mentioned before, this parameter is configurable depending on the selected goal, either to increase number of objects (wider area covered) or to improve the accuracy of part of the objects (at least 2 re-observations).

Finally, contrary to the leak-proof concept, they could exist cases (i.e. telescope with a very wide FoV) where the whole stripe can be swept in a shorter period of time than the objects use to cross the FoV. For those cases, the cost function is also subject to an additional constraint in order to try to reduce the number of observed regions of the sky:

$$\alpha_{corr} = t_{\mathrm{stripe}} n_{\mathrm{GEO}} - FoV + \gamma_r \quad (3)$$

Where:

- α_{corr} is the correction in right ascension between stripes
- t_{stripe} is the time spent in the stripe ($t_{\mathrm{stripe}} = t_{\mathrm{exp}} N_i + t_{\mathrm{rep}}$)

A more detailed explanation of this approximation and the obtained results can be found in [1].

2.2 Survey visibilities

Once the telescope pointing ephemerides have been defined based on the survey strategy and fields of regard of the radars have been configured, all the survey visibilities can be computed using the complete catalogue of objects.

For telescope observability, constraints described in (1) are considered in addition with two more detectability constraints:

$$\omega_{\text{obj}} < \omega_{\max} \tag{4}$$

Where:

- ω_{obj} is the angular velocity of the observed object
- $\omega_{\max}$ is the maximum angular velocity a sensor can follow

and

$$m = m_0 - 2.5\log_{10}\left(\frac{\frac{2}{3\pi^2}F_S\frac{\pi d_s^2}{4R^2}a(\sin\theta + (\pi-\theta)\cos\theta)}{F_0}\right) < m_{lim} \tag{5}$$

Where,

- m and m_0 are the apparent magnitude of the object and of the reference object, respectively (for Sun $m_0 = -26.74\, mag$).
- F_S and F_0 are the irradiance flux of the Sun and the reference object (Sun), respectively.
- d_s is the diameter of the object, assumed a Lambertian sphere for simplicity.
- R is the range between the telescope and object.
- a is the geometric albedo of the object (assumed to be 0.1 for all objects).
- $m_{lim}$is the limiting magnitude of the telescope

For radar observability the only condition that applies is the minimum elevation whereas for the detectability angular restriction applies, (4), together with the so call "radar equation", defined as follows:

$$RCS_{\text{obj}} > \text{RCS}_{\min} = \frac{RCS_{ref}\rho_{obj}^4}{\rho_{ref}^4} \tag{6}$$

Where:

- RCS_{obj} and ρ_{obj} are the Radar Cross Section (assumed the area) and range of the observed object
- RCS_{ref} and ρ_{ref} are the Radar Cross Section (assumed the area) and range of reference

After applying these constraints, a visibility for a sensor-object pair is obtained when the object is inside the FoV, so that:

$$Conical\ FoV: \bar{x}_{point} \cdot \bar{x}_{object} \leq \frac{FoV}{2}$$

$$Pyramidal\ FoV: \begin{cases} a = \bar{x}_{object} \cdot (\bar{x}_{point_{+x,+y}} \times \bar{x}_{point_{+x,-y}}) \\ b = \bar{x}_{object} \cdot (\bar{x}_{point_{+x,-y}} \times \bar{x}_{point_{-x,-y}}) \\ c = \bar{x}_{object} \cdot (\bar{x}_{point_{-x,-y}} \times \bar{x}_{point_{-x,+y}}) \\ d = \bar{x}_{object} \cdot (\bar{x}_{point_{-x,+y}} \times \bar{x}_{point_{+x,+y}}) \end{cases} satisfying \begin{cases} a,b,c,d \geq 0 \\ a,b,c,d \leq 0 \end{cases} \tag{7}$$

2.3 **Survey accuracy gain**

The main objective of the survey chain is to compute the covariance improvement of each of the objects observed by survey sensors in order to use it as a feedback for the tracking chain.

The object's initial covariance (P_0) is retrieved from the state of the object. This covariance can be propagated to any time using the transition matrix as:

$$P_i = \Phi_{t_i,t_0} P_0 \Phi_{t_i,t_0}^T \tag{8}$$

Where, $P_i = P(t_i)$ represents the statistical covariance at time t_i and $\Phi_{t_i,t_0} = \Phi(t_i, t_0)$ the transition matrix between states t_0 and t_i. Physically it represents the evolution of errors in time, that is, how an error in a certain component in the reference state is transformed into an error in the same or any other component at the start epoch of the analysis.

With the visibility periods obtained during the survey, a block of measures can be simulated and its effect on the covariance evaluated as follows:

$$\hat{P}_i^{-1} = P_i^{-1} + G_i^T W G_i \; where \; G_i = \frac{\partial \overline{z_i}}{\partial \overline{x_i}} \tag{9}$$

Where $\hat{P}_i$ represents the covariance updated by the measurements at time t_i, P_i the covariance propagated at the time of the measurements, G_i the matrix of partial derivatives of the measurements ($\overline{z_i}$) with respect to the state vector ($\overline{x_i}$) (physically it translates the error in the measurement into position and velocity errors of the object) and $W = diag(\sigma_\beta^{-2}, \sigma_\varepsilon^{-2}, \ldots, \sigma_{\dot{\rho}}^{-2})$ represents the weight of the different components of the measure. This weight is inversely associated with the expected error of each measurement component, weighing the components between them and therefore their effect on the state vector.

.

The effect of each block of measurements into the state vector of the object can also be propagated to any time applying the same transition matrix:

$$H_i = \frac{\partial \overline{z_i}}{\partial \bar{x}_f} = \frac{\partial \overline{z_i}}{\partial \overline{x_i}} \frac{\partial \overline{x_i}}{\partial \bar{x}_f} = G_i \Phi_{t_i,t_f} \tag{10}$$

And therefore, the effect of the block of measurements in the covariance at the end of the computation period is:

$$\hat{P}_f^{-1} = P_f^{-1} + \Phi_{t_i,t_f}^T G_i^T W G_i \Phi_{t_i,t_f} = P_f^{-1} + H_i^T W H_i \tag{11}$$

Where $\Phi_{t_i,t_f} = \Phi(t_i, t_f)$ is the transition matrix that leads from the point where the covariance is to be evaluated to the package of the considered measurements. Adding up all the packages of measurements and considering the propagation of the reference covariance, the equation would be as follows:

$$\hat{P}_f^{-1} = \left(\Phi_{t_f,t_0} P_0 \Phi_{t_f,t_0}^T\right)^{-1} + \sum_{i=1}^{n} H_i^T W H_i \tag{12}$$

Notice the nature of two terms in above equation. First represents the effect on the final covariance of the propagated initial covariance and the second one the effect of considering the survey observations into the covariance.

The following figure is an example of the accuracy gain concept.

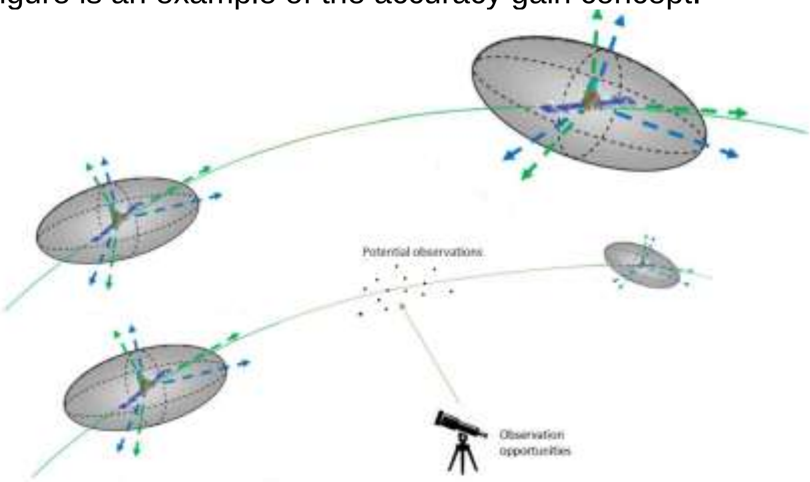

Figure 3: Accuracy gain obtained with potential observations from a sensor

In the upper part of the figure, the natural evolution of the uncertainty of the orbit tends to increase with time, making the covariance ellipsoid become greater with time. However, in the lower part of the figure, it is illustrated what happens if observations are included in the orbit determination process. The resulting covariance after the orbit determination for the same epoch as in the upper example is considerably reduced by the effect of the observations, increasing the accuracy.
Applying this process, it is possible to estimate the effect of the survey strategies over the accuracy of the overall objects, and therefore, considering it also in the tracking chain to optimize the follow-up of the objects that do not reach the targeted accuracy.

This formulation is based on the nomenclature of classic references such as [2] and [3].

3 TRACKING CHAIN

Tracking chain is used by telescope, radar and lasers stations under tracking activity. This chain is mainly dedicated to objects under special requests from the SST services, high degraded accuracy in the catalogue, calibration campaigns and dedicated exercises. In order to extract optimal schedule of the sensor network, this chain should be executed after the survey considering the observations likely be provided by the survey sensors and their effect on the objects.

3.1 Object characterization

The starting point of the tracking chain is the object characterization whose main functions are the classification of the objects in different groups, known as "population groups" and the establishment of the priorities for the full object's catalogue that will be used later for the optimizer.

Population groups are defined by the user according to object properties such as, status of the object (debris, active and unknown), orbital regime (LEO, MEO, GEO and GTO), event source (collision avoidance, re-entry, fragmentation, calibration and others). Groups have also an associated priority multiplier that has been configured according to COPLA internal priorities. Final priority of each object is computed in the following way:

$$F = \mathrm{f}_1\mathrm{f}_2\mathrm{f}_3 = \mathrm{f}_1\frac{1}{2}\left(\frac{C}{W_c C} + \frac{t_{last\ obs}}{W_{last\ obs}t_{last\ obs}}\right)\frac{1}{2}\left(\frac{A}{W_A A}\right) \quad (13)$$

Where:

- f_1 is the priority multiplier for each population group
- f_2 is the accuracy factor, composed of C (main diagonal of the updated covariance of the object, (12)), W_c (weight for covariance), $t_{last\ obs}$ (time to last observation, acquired from the last measurement obtained of the given object) and $W_{last\ obs}$ (weight for last observation)
- f_3 is the size factor, composed of A (area of the object) and W_A (weight of the area)

Notice that this priority allows the COPLA operator to fine-tune three different weights in order to give more weight to one of the mentioned factors. Notice also the feedback of the survey chain in the priority computation.

Additionally, there exists another source of priority, the EUSST Tasking Request for SST services. When a tasking request exists in COPLA, the above priority is computed in the following way:

$$F = \mathrm{f}_1 f_{TR} \quad (14)$$

Therefore, according to this equation, the priority of the object only depends on its internal multiplier and the priority assigned to the EUSST Tasking Request, stablished at Consortium level.

3.2 Tracking opportunities

Tracking opportunities are computed in a similar approach as it was explained in Section 2.2. The only difference is that for the case of opportunities the condition of having the object inside the FoV is changed by the Field of Regard (FoR). FoR can be defined as the total area a sensor can cover by changing its pointing. This condition is implicitly met with the observability constraints.

COPLA software will also allow a constraint relaxation in this first phase, so that its main objective is just to obtain a set of pre-filtered opportunities with the most restrictive or physical conditions. In a second phase, a probabilistic approach will be applied on those opportunities. The probabilistic approach allows considering the uncertainty in external or environmental factors, and relationships between the different constraints, simply giving a value of the joint observation probability of a given object for each moment. For this approach, machine learning algorithms will be used since they can implement simple neural networks that are capable of representing the effect on the probability of detection as a function of the observation parameters, elevation, separation and phase of the moon, brightness, etc ... Although these models can be trained by simulated data considering the current restrictions, this implementation would also allow re-training based on the history of observation plans and the measurements made (Figure 4-Right), helping to consider the initial lack of definition of said restrictions or to consider cross effects. In case the history is not available, data can be simulated using the same cut-off functions used in the deterministic approach and allow machine learning to smooth these cuts (Figure 4-Left).

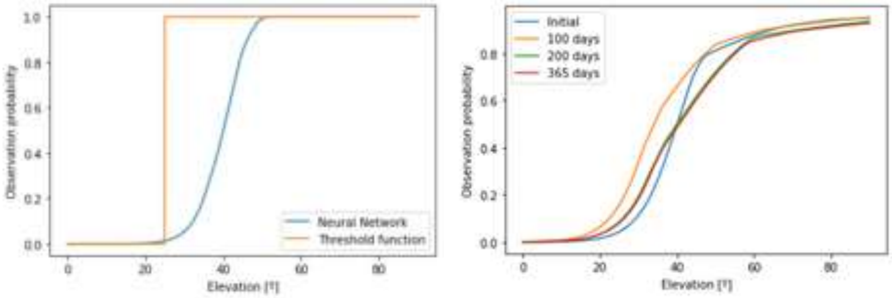

Figure 4: Statistical calculation of visibility and retraining opportunity

As indicated before, machine learning would allow the algorithms to train unmodelled relationships between observation conditions. These relationships could be very complex to be quantified even by the sensor operator, but it is expected the ML algorithm to extract the information need based on the historical data. An example of these relationships could be the observation probability of an object depending on the separation with respect to the moon, where the brightness of the object, the phase of the moon and even the elevation of the object could simultaneously affect the constraint.

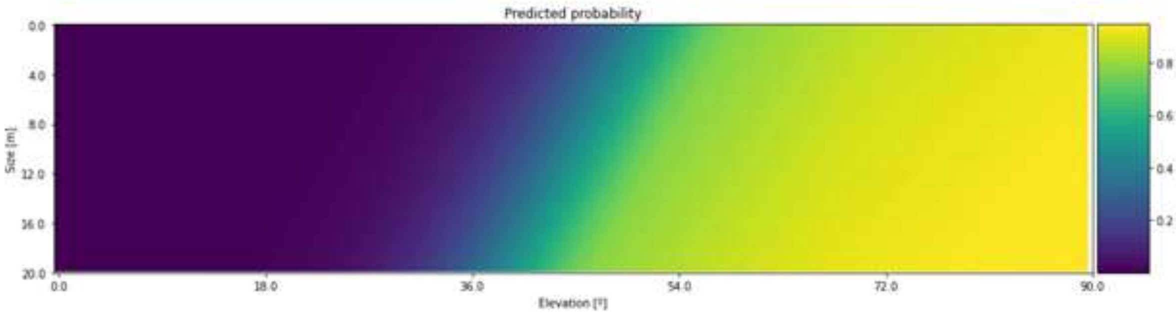

Figure 5: Combined elevation and object size constraint

Summarizing, the combination of the relaxed constraints and the probabilistic approach would allow obtaining a reduced set of slots with an associated “observation probability” that will be later used by the optimizer as part of the cost function.

3.3 **Sensor performances weighting**

An additional contribution to the schedule optimisation is given by the sensor performance indicator. A model based on the history of previous plans and the characteristics of the sensor, and the objects are used to estimate the success probability that a given slot will be actually observed by the sensor. This success probability is used by the planning algorithm as an additional contribution to the cost function of each slot. Two kinds of models are available for computing the sensor weight: one is a simple statistics-based model, based on the success rate of previously planned slots to that sensor in the past.

The second model is a Machine Learning algorithm, which uses more information about the object (such as elevation, azimuth, time of the slot, angular distance to the moon in the sky, etc.) to provide a better ad-hoc estimation for the slot. This model is trained from previous planned slots, and it is only available when a significant number of samples are available. In summary, it takes into consideration information about the slot, the target object properties, trajectory and observation conditions.

This machine learning model uses a Random Forest algorithm, which uses a set of decision trees, each one built from a sample drawn with replacement, and at each node the best split is found using information of a subset of features or samples. Then the prediction of all trees is averaged for the result. Thus, this algorithm is part of the ensemble learning family. Random Forest was selected as the best-performant algorithm of several tested, including neural networks, other decision-tree-based algorithms and different regression models, from the results of [4].

The model was tuned and trained using a large dataset of historical plans, tracks and reported summary of the nights from the S3TOC database, with additional data augmentation to include orbital information (missing in the source data). The dataset contains 1200 observation requests with more than 150.000 observation slots (target object, time slot, sensor, sensor location) for a time period between May 2017 and June 2020.

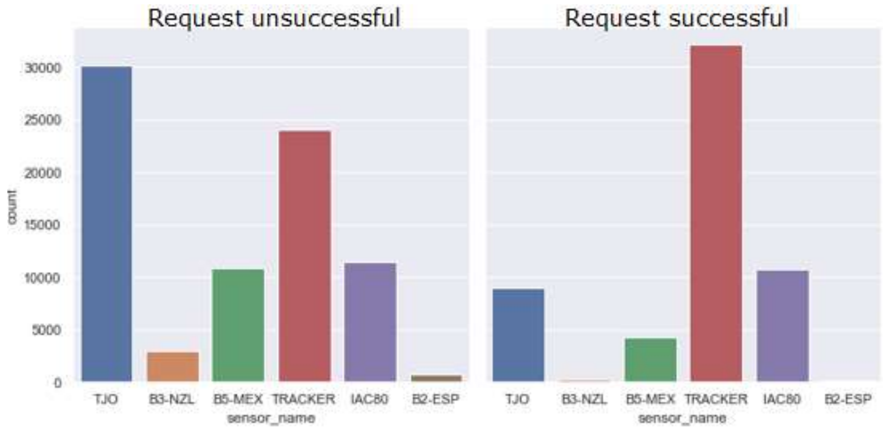

Figure 6: Distribution of observation requests per sensor

Special care was used to maximize the prediction accuracy of the model without overfitting by tuning the model's hyper-parameters, so the model from past observations can be generalized to future predictions, although the sensor weighing software can re-train the model at any time.

Three hyper-parameters are tuned to reduce the overfitting: the number of estimators, the maximum depth of the decision trees and the percentage of the total number of samples to consider for node splitting.

The final model Confusion matrix is shown in Figure 7, which shows that for successful requests, the precision is around 84% (true positives rate) whereas for unsuccessful requests is about 68% (true negatives rate).

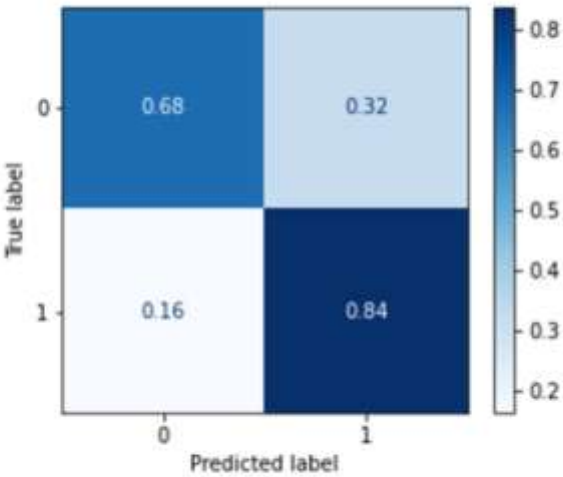

Figure 7: Confusion matrix for the final Random Forest Model on the test set in [4]

The false positive rate is about 32%, which could be decreased to make the model more accurate and effective. However, it is better to overestimate the probability of success than underestimate and lose opportunities to observe objects.

3.4 **Tracking accuracy gain**

Last performance indicator for slot selection is the accuracy gain. The algorithms used for tracking accuracy gain are the same than the ones explained in 2.3., with the only difference that in this case the initial covariance to which they are added the slot improvement will be the updated one after the survey campaign, i.e. $\hat{P}_f^{-1}$ defined in (12).

3.5 **Tracking plan optimization**

A solver is a mechanism to implement constraint programing to achieve the best solution of a problem. In constraint programing, optimization is done by computing improving solutions, until reaching an optimum.

The problem of selecting the appropriate objects to be observed by the sensor can be treated as a constraint's satisfaction problem. Taking into account the provided input, the opportunities divided into slots, the solver should select the object to observe for the pair sensor and slot according to the opportunity conditions.

In that way, it is possible to constraint for a minimum number of slots per sensor and assure the sensor will observe the same object during the stablished minimum period, for example, if the sensor needs a minimum of 6 minutes to observe one object and the default slot is 1 minute, the solver should assure there will be 6 consecutive slots of the interval with the same object assigned to the sensor. This constraint is defined as hard constraint, i.e., the solver has to meet it to provide a feasible solution. On the other hand, the soft constraints can be defined to optimize the feasible solution.

The soft constraints will try to balance between the quality of the observations and the movements of the sensors to observe the objects. It can be defined as the cost function with the following formula:

$$\sum_{j=0}^{obj} \frac{1}{F_{obj}} \cdot \hat{P}_{fT} \cdot w_1 + \sum_{s=1}^{sens} \sum_{m=0}^{mov} \alpha_m \cdot w_s \quad (15)$$

$$\hat{P}_{fT}^{-1} = \hat{P}_{fS}^{-1} + \sum_{i=0}^{obs\,T} H_i^T W H_i \cdot F_{obs}(1 - F_{cc}) \cdot F_{sw} \quad (16)$$

Where:

- F_{obj} represents the priority of the object, the objects with the highest priority weigh have preference to be observed.
- $\hat{P}_{fT}$ represents the final orbital accuracy gain with respect to the updated covariance from survey operations ($\hat{P}_{fS}$) considering all the observations of the selected slots of the same object for tracking $\sum_{i=0}^{obs\,T} H_i^T W H_i \cdot F_{obs}(1 - F_{cc}) \cdot F_{sw}$.
- F_{obs} represents the probability of success of the slot according to the probability estimated by the opportunity calculation.
- F_{cc} represents the forecasted probability of cloud cover.
- F_{sw} represents the figure of merit calculated based on the historical performance of the sensor, i.e. sensor weight.
- α_m represents the angle of by the sensor for the movement from slot m to slot $m + 1$.
- w_1 and w_s correspond to configurable weights by the COPLA operator that will allow adjusting the relative weight of the observation quality and the movement of the sensors.

The objective can be summarized as the minimization of the uncertainty of the overall catalogue, minimizing the average covariance weighted with each object priority and, at the same time, the minimization of the total movement of each sensor that spends observation time and increases the wear of the frames. It is noted that both contributions can be tuned through a good selection of the relative weights (w_1 and w_s).

The solver tries to find the optimal solution exploring the search space, presented as a tree where each node is a solution. The movement from one solution to another has been modelled as the change in the value of the observed object for the pair slot/sensor. Per movement, the score function is evaluated and depending on the selected search strategy the next branch of the tree is explored up to a certain termination criteria, such as, for example, the limit number of nodes to be explored has been reached, the cost function has reached the desired value, the solution found is the best after k iterations, the execution time limit has been achieved, etc. In this case, the termination criteria is defined as a limit in the execution time. In the performed simulations, the execution time limit is set to 15 minutes.

In the solver optimization process, the following phases can be configured:

- The Construction Heuristic (CH) initializes the values of the planning variables according to the score (hard/soft constraints) and the selected strategy.
- The Local Search (LS) works from an initialized solution that evolves during the search according to the score and the chosen search algorithm.

In order to easily define the problem, the constraints and the search strategy, the solver OptaPlanner (http://www.optaplanner.org) is used. This engine combines optimization heuristics and metaheuristics with a very efficient score calculation. The metaheuristics algorithms are a kind of stochastic optimization algorithms, which uses some degree of randomness to find the best solution for NP-hard problems and it is proven they have better performance than other techniques. To implement the score calculation the problem constraints are defined as business rules and assigning a score to each one, making it easy to implement and scale.

The search strategy and the termination criteria can be easily configured in OptaPlanner and provides the possibility to benchmark several configurations in order to find the one that better fits with the specific problem data. An algorithm that checks every possible solution (even with pruning, such as in Branch and Bound) can easily run for billions of years on a single real-life planning problem. The aim is to find the best solution in the available timeframe. Planning competitions (such as the International Timetabling Competition) show that Local Search variations (Tabu Search, Simulated Annealing, Late Acceptance…) usually perform best for real-world problems given real-world time limitations.

For this specific problem, using a dataset with 4 sensors, 30 objects and an interval of 1 day, with opportunities calculated for each 5 minutes, several simulations are run. The selected CH is First Fit strategy and for the LS several configurations are evaluated.

In the first simulation, the configuration of Tabu Search strategy has been evaluated. The parameters to configure in this algorithm are the Tabu size and the accepted count limit (according to OptaPlanner documentation this value should be high). Testing reveals that the best values for the Tabu Search configuration are 0.2% for the ratio and 10000 for the accepted count limit, as it can be seen in the following table:

Table 1: Tabu Search simulation results

LS Configuration	**Final Score**	**Raking**
TS_Standard	[0]hard/[-286440/12124]soft	4
TS_size7_limit100	[0]hard/[-618172/11856]soft	5
TS_size7_limit1000	[0]hard/[-276448/12187]soft	2
TS_size7_limit10000	[0]hard/[-262866/12369]soft	1
TS_ratio2_limit100	[0]hard/[-648373/12057]soft	6
TS_ratio2_limit1000	[0]hard/[-285689/12159]soft	3
TS_ratio2_limit10000	[0]hard/[-261653/12362]soft	0

The score is defined with, in the hard level the constraint of the minimum number of slots per sensor to observe an object and in the soft level, first value is the movement of the sensor and the second value is the uncertainty of the objects.

Similar simulations have been performed for the Simulated Annealing algorithm. In that case, the configured parameters are the accepted count limit and the starting temperature. The best results are with the configuration of starting temperature of [2000]hard/[1000/100]soft and accepted count limit of 4 (see Table 2).

Table 2: Simulated Annealing simulation results

LS Configuration	**Final Score**	**Ranking**

SA_2000hard_20000_100soft_a4	[0]hard/[-206706/369]soft	2
SA_2000hard_100000_100soft_a4	[0]hard/[-206706/369]soft	2
SA_2000hard_100000_10soft_a4	[0]hard/[-206706/369]soft	2
SA_1000hard_10000_100soft_a4	[0]hard/[-206444/360]soft	1
SA_1000hard_10000_10soft_a4	[0]hard/[-206706/369]soft	2
SA_2000hard_1000_100soft_a4	[0]hard/[-190128/364]soft	0

A third benchmark simulation is executed with several LS algorithms and the Tabu Search is the best for the soft score level 0, while Simulated Annealing is the best for the soft score level 1, as shown in the following plots. The first plot is how the score of sensor movements evolve during time and the second plot is with the score of the uncertainty of the objects.

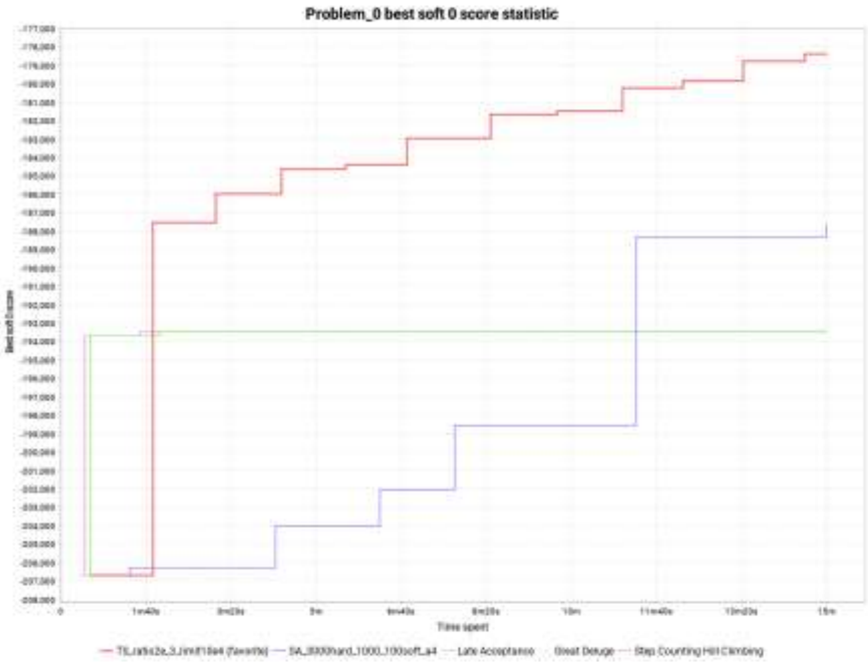

Figure 8: Benchmark simulation for sensor movement in OptaPlanner

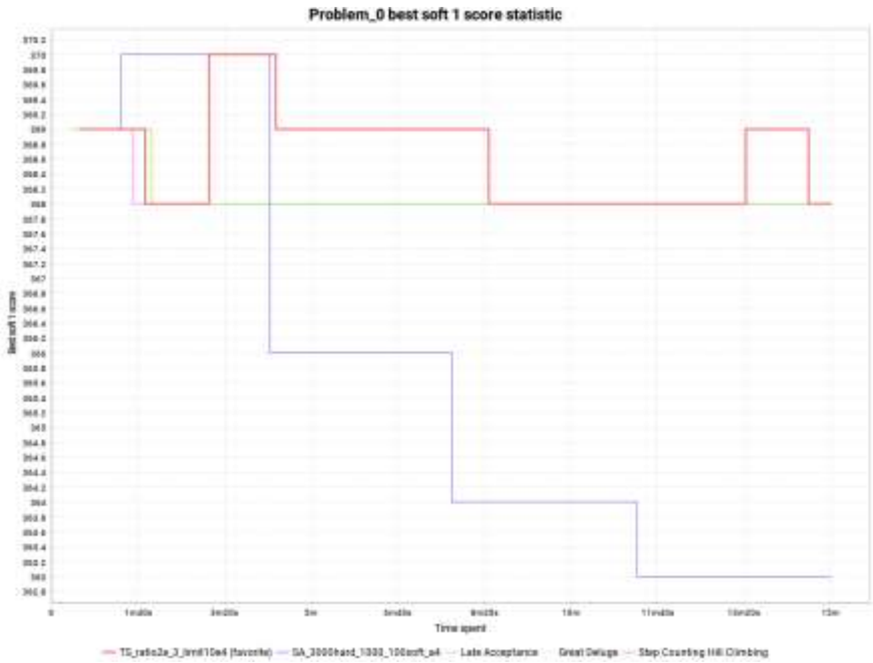

Figure 9: Benchmark simulation for object uncertainty in OptaPlanner

4 CONCLUSIONS

This paper has explained, in a detailed way, all the implemented algorithms for COPLA, whose main goal is the cooperation among all the EUSST sensors for providing accurate SST services.

Firstly, for the survey chain, a parametrization on the free parameters of the survey strategy for telescopes is performed. This would allow to obtain optimal results in terms of survey time and track success (strictly related to the brightness of the objects). The other key point of this chain, is the covariance feedback obtained that will be inserted in the tracking chain, thus avoiding excess or defect of tracking slots assigned by the combination of both chains.

Secondly, for the tracking chain, a prior selection and classification of the objects is done in order to narrow down the problem by extracting all the information regarding SST services and the survey chain for the objects. Artificial intelligence is also used for the better evaluation of the sensor-object opportunities, i.e., cross components, smooth behaviours and historic results for sensors or objects. Finally, the optimization solver returns the optimal tracking observations, known as plan slots, considering the accuracy gain, the object priority, the observation probability and the sensor performance weight together with sensor movement penalization in the specified execution time.

5 REFERENCES

[1] D. Sáez-Bo, D. Moreno, A. Pastor, E. Villanueva and D. Escobar, “Radar and optical strategies characterization and optimization,” in *2nd IAA Conference on Space Situational Awareness (ICSSA)*, 2020.

[2] G. Montenbruck, Satellite Orbits: models, methods and applications, 2000.

[3] D. Vallado, Fundamentals of astrodynamics, 2013.

[4] I. Urdampilleta, D. Lubián, I. Grande and F. Pina, “Machine learning applied to the optimization of SST sensor tasking,” in *8th European Conference on Space Debris*, 2021.