A Mixed Integer Linear Programming Model for Resolution of the Antenna-Satellite Scheduling Problem

LORENA LINARES
RAFAEL VAZQUEZ, Senior Member, IEEE
FEDERICO PEREA
JORGE GALÁN-VIOQUE
University of Sevilla, Sevilla, Spain

This article deals with one of the types of "Satellite Range Scheduling" problems arising in Earth Observation Satellite operations, Antenna-Satellite Scheduling. Given a set of satellites, a set of available antennas and a time horizon, the problem consists of designing an operational plan that assigns satellites to antennas in an optimal fashion. Extending a previous integer linear programming (ILP) model (*shortening model*, with only integer variables), we propose a mixed ILP (MILP) (*shaving model*, which includes both continuous and integer variables), to more efficiently solve this problem. After computing the passes generated by the satellites' windows of visibility from the antennas, the optimal planner is able to cancel a pass, move it to another antenna, or shorten its duration, in order to avoid scheduling conflicts. In contrast to the shortening model, which used intersections between passes to determine the best schedule, the shortening operation is now referred to as shaving, since the shaving model can arbitrarily adjust the duration of a pass in a razor-like fashion, giving the model its name. Computational results obtained in tests over realistic scenarios prove that the shaving model outperforms the shortening model, producing fewer cancellations, smaller shaved times, and a *fairer* distribution of cancelled passes among satellites, with much shorter preprocessing times.

I. INTRODUCTION

Earth observation satellites (EOSs) have experienced a considerable growth in the last decades. These satellites generate very large datasets, and, consequently, require efficient communication with ground antennas to transmit their acquired information. The problem of scheduling downlink (or uplink) connections between satellites and antennas falls under the umbrella of the "Satellite Range Scheduling" (SRS) problem, which has been shown to be NP-complete [1], [2]. When many satellites and antennas are considered, the problem therefore becomes intractable and requires the use of advanced algorithms to obtain good solutions in short computing times.

Numerous researchers have proposed solutions to the SRS problem in the last two decades. The first seminal results were reported by Barbulescu et al. In [3] the SRS was solved, applied to the US Air Force Satellite Control Network (AFSCN), which had at that time over 100 spacecraft, 16 antennas, nine stations, resulting in about 500 requests per day. Finding typically over a hundred conflicts, genetic algorithms performed better than other algorithms. In a posterior work, the SRS problem was analyzed and found to be formally NP-complete [1], [2]. New algorithms are provided, improving previous results. Next [4] gave an overview of how the problem was addressed during the previous ten years at the AFSCN. In addition, they explored different cost functions, for instance one that reduces overlaps. Finally, other heuristic algorithms for SRS were proposed [5], including combinations of the previous ones.

These initial publications were soon followed by works from other groups. To cite a few, another work explored the SRS problem for the deep space network (DSN) [6]; the corresponding scenarios have 16 antennas, 20 satellites, a horizon of four months, which results in over 500 passes each week. They generated and repaired schedules, and proposed heuristics for solving the problem with emphasis in rescheduling. In another approach, the authors integrated genetic algorithms, graph theory and linear programming in order to build conflict-free plans [7], and applied their approach to a practical case study provided by a satellite service company. Other authors studied the scheduling of a single geostationary satellite [8]. In another example, the problem was formulated as a mixed integer linear programming (MILP) model, solved by means of a Lagrangian relaxation [9]. As a case study, they applied their approach to

Manuscript received 24 January 2023; revised 7 August 2023; accepted 10 October 2023. Date of publication 23 October 2023; date of current version 9 February 2024.

DOI. No. 10.1109/TAES.2023.3326422

Refereeing of this contribution was handled by G. Chen.

The work of Rafael Vazquez was supported in part by MCIN/AEI/10.13039/501100011033 under Grant TED2021-132099B-C33 and in part by European Union NextGenerationEU/PRTR. The work of Federico Perea was supported in part by the Spanish Agencia Estatal de investigacion (AEI) and in part by the European Regional Development's fund (ERDF) under Grants PID2020-114594GB-C21 and PID2021-124975OBI00, and in part by the Regional Gopvernment of Andalusla under Projects FEDERUS-1256951 and AT21_00032. The work of Jorge Galán Vioque was supported by the Ministerio de Clencia e innovacion under Grants PID2021-123200NB-I00.

Authors' addresses: Lorena Linares and Rafael Vazquez are with the Department of Aerospace Engineering, Fluids Mechanics, ETSI, University of Sevilla, 41092 Sevilla, Spain, E-mail: (llinares@us.es; rvazquez1@us.es); Federico Perea is with the Applied Mathematics II, EPS, University of Sevilla, 41004 Sevilla, Spain E-mail: (perea@us.es); Jorge Galán-Vioque is with the Applied Mathematics II, ETSI, University of Sevilla, 41092 Sevilla, Spain, E-mail: (jgv@us.es). *(Corresponding author: Jorge Galán-Vioque.)*

Galileo. In [10], the authors solved SRS by using struggle genetic algorithms on STK simulations. In another work ant-colony algorithms were proposed [11], solving examples with 17 satellites and 11 to 13 antennas, yielding around 400 passes. To cite another approach, graph coloring algorithms have also been applied [12], with examples of up to 500 realistic instances. Finally, a more global point of view was adopted in [13], where the authors try to integrate automated scheduling into the concept of timeline (a track record of spacecraft states and resources).

This problem has also arisen in the context of academic ground station networks [14], [15] for small satellites operated by research institutions, which usually have some specific needs such as redundancy and flexibility. Similarly, this problem has been solved with a tailored approach that also maximizes redundancy in order to solve possible failures in communication [16], where a simple scenario with six satellites and four stations is considered, yielding 51 contact windows.

Some other more recent results are explored next. For instance, Chen et al. [17] also used a MILP approach to solve the problem of multisatellite scheduling. The case problem has limited observation capacities, so stronger constraints were used to speed up the solution process. It combined three satellites and different target sets to generate 37 problem instances with a scheduling horizon of 24 or 48 h. Due to its NP-completeness, heuristic methods have been widely adopted when solving the SRS problem. In [18], the Agile Earth Observing Satellite (AEOS) scheduling problem was solved by proposing a temporal conflict network-based heuristic algorithm with a satellite and 15 generated scenarios. These scenarios contained targets that varies from 50 to 400 in a 24 h scheduling time horizon. A detailed summary of the AEOSs problem was given in [19]. There were two approaches to its formulation, the discrete time model and the continuous time model.

It is usual to find academic articles developing different deterministic algorithms along with their heuristic analog, as in [20]. The article introduces a deterministic optimization model for downloading large data quantities, and develops a scheduling heuristic that mimics a traditional process to compare the results. It generates a real-world space mission consisting of 50 satellites and 50 ground stations in a three-month time span. A notorious example is [21], which designs a MILP formulation of the NASA's DSN scheduling problem, and later a non-MILP based heuristic to validate the results for a real week with 14 resources, 286 activities, 1430 h of tracking time, and 30 missions. Subsequently, this same formulation is improved in [22], which introduces a new set of constraints and compared the results with the previous one. The variant allows us to prioritize emergency or landing scenarios and satisfies all the requested constraints. Another MILP formulation is proposed in [23] for optimal scheduling of AEOSs, but it requires such a long computational time, which a heuristic algorithm is needed to find a near-optimal solution. The approach generates a case study of four satellites and eight problem instances with 50–100 observations tasks and random ground stations.

In addition, it is important to highlight the progress in solving the SRS problem thanks to the flourish of genetic algorithms, deep learning and evolutionary algorithms. There is a wealth of relevant papers developing these artificial intelligence techniques. For instance, genetic algorithms are used in [24] with a prominent superiority solving large-scale problems with a task size that varies from 25 up to 400 tasks. The improved genetic algorithm for the ground tracking telemetry and command tasks in multisatellite mission scheduling with population perturbation and elimination, is compared with a genetic algorithm, an adaptive neighborhood local search algorithm, and a local search one, upgrading the performance in profit and task completion rate. A multirelay satellite resource scheduling algorithm based on ant colony genetic algorithm is proposed in [25]. This algorithm has a fast convergence speed and a strong global optimization ability solving multiconstraint conditions. The simulation scenario consists of 6 h span with 64 tasks, three satellites and two antennas for each one. Deep learning and heuristics are combined in [26]. It trains a method using three satellites observing scheduling results for one day, and gives a high-quality solution for the 5350 experimental scenarios. Another example is [27], where the classical technology and machine learning are integrated with a Support Vector Machine classification (SVM) approach, combines with a multiobjective evolutionary algorithm (NSGA-II). The model solves 546 tasks in six simulation scenarios with five ground stations (two antennas for each one) and from 10 to 60 satellites for a day planning horizon. In the last year, Zhang and Xing [28] formulated a MILP model and develops an improved genetic algorithm with a novel idea of encoding and decoding to match the specific request with the corresponding satellite-ground resources. It establishes five scenarios with five satellites and five ground stations for a day span. Finally, Wang et al. [29] dealt with the multiple EOSs scheduling problem by formulating it as an unrelated parallel machine scheduling problem with multiple time windows. The article takes into account observation targets and satellites as jobs and machines and develops two mixed-integer linear programming formulations with additional inequalities to refine the formulation. It solves 47 problem instances, with 100 up to 1200 number of jobs in a scheduling horizon setting of one day.

As a summary, the most relevant contributions of recent years with their main features is given in Table I.

A. Contribution and Novelty With Respect to Previous Work

From the most recent literature, we can see that solving the antenna-satellite scheduling problem in large scenarios with numerous passes is still relevant today and no single algorithm has emerged as the most efficient approach. Vazquez [30] was our first work on this problem; a discrete shortening model (which only allowed to shorten the passes to certain values) was solved by an ILP. Inspired

TABLE I
Summary of Recent Works on the Antenna-Satellite Scheduling Problem

Reference	Optimization Method	Description	Scenarios
Chen et al. [26]	Machine learning-based hierarchical prediction model and heuristic local search algorithm with Python and CPLEX	Maximizes observation tasks with their priorities	5350 experimental scenarios 1000 observation targets 3 satellites 1 day span
Zhang and Xing [28]	Genetic Algorithm with MATLAB	Maximizes requests hosting	5 scenarios 6 satellites and 5 antennas 1 day span
Claudet et al. [22]	An upgraded MILP with MATLAB and GUROBI	Maximizes the number of tracks (or scheduled view periods) and activities	14 resources (284 activities) 1418 hours of requested tracking time 1 week span
Zhao et al. [25]	Ant colony algorithm and Genetic Algorithm	Maximizes task priority	3 satellites 2 antennas for each satellite 6 hours span with 64 tasks
Kim et al. [23]	MILP formulation with Python and GUROBI	Maximizes selected task	4 satellites 50-100 requests in different antennas
Chen et al. [24]	Genetic Algorithm with MATLAB	Maximizes execution possibility and completed tasks	7 scenarios with 25-400 tasks Numerous satellites and antennas
Chen et al. [17]	MILP formulation with GUROBI	Maximizes requests hosting or completed tasks	37 problem instances 3 satellites 1 or 2 days span
Xie et al. [18]	Heuristic with C#	Maximizes observations benefit	15 scenarios 1 satellite 1 day span
Lemay et al. [20]	Deterministic system with C++ and CPLEX	Maximizes the amount of information and its efficiency	50 satellites 50 antennas 3 months span (1 minute resolution time)
Vazquez et al. [30]	ILP model in MATLAB and LPSolve	Maximizes the priorities and the active time of passes	10 real instances 50 satellites and 3000 passes 20 antennas A week span

by a result in a closely related area (namely, optimization of swath acquisitions, see, e.g., [31]), one of the main ideas was computing all self-intersections produced by the set of intervals of the passes (subpasses), which gives the discrete values which the passes can be shortened to. This approach solved the conflicts respecting the priorities and preferred assignments of ten real instances, containing around 3000 passes requested in a week, with 50 satellites and 20 antennas. These results are compared in this article against our approach. Next, in the conference paper [32], we preliminary proposed an improved optimization algorithm based on MILP to reduce the number of cancellations, solving three scenarios of up to 18 satellites and six antennas with 1790 passes, in a time frame of one day and one week. In this work, shortening was now referred to as shaving, since it can arbitrarily adjust the duration of a pass in a razor-like fashion thanks to the inclusion of new continuous variables, in contrast with our previous ILP formulation which only allowed integer variables.

We have developed a new shaving formulation in this article, which now avoids computing the subpasses and reduces drastically the preprocessing time. We achieve feasible solutions for large scenarios with thousands of passes, attaining at the same time fewer cancelled passes (with a *fairer* distribution among satellites). In addition, it is proven that missed times (when satellites do not connect to antennas for the full duration of a pass due to another satellite already being connected) in the shaving model, referred to as shaved times, are significantly lower, compared with the previously shortened times in the shortening model. All these facts make our model a suitable alternative to deal with the antenna-satellite scheduling problem.

B. Structure of the Article

The rest of this article is structured as follows. After the present introduction, Section II states the antenna-satellite scheduling problem and establishes some notation used throughout the article and the initial processing of data required for our algorithm including the computation of passes from the most basic data (satellites, antennas, and a time frame). Next, Section III is devoted to the shaving model, giving details on the variables, constraints, and objective functions used in it. We continue in Section IV by specifying 32 scenarios used to perform numerical experiments, and give the results comparing the shaving model with our previous shortening model. Finally, Section V concludes this article.

II. PROBLEM SETTING AND COMPUTATION OF PASSES

This section formulates the initial data for the antenna-satellite scheduling problem and states our goal; then, it explains how to process these inputs to obtain the inputs required for the shaving model. Next, a brief example of how conflicts are managed is explained, and finally, a list of all the used satellites is presented.

A. Problem Input

The initial data of the antenna-satellite scheduling problem are as follows:

1) The time horizon for the scheduling problem, which is an interval $T = [T_i, T_f]$ with the initial and final times, respectively.
2) A set of antennas A with their geographical coordinates, as well as their visibility masks. In our example scenarios, we consider for simplicity antennas located at the same site.
3) A set of satellites N, with their orbital data in, e.g., TLE format. See Section II-B. Each satellite generates *passes* when flying over the antennas. Let P be the set of all passes.
4) A binary valued matrix C with $C_{jk} = 1$ if satellite j is compatible with antenna k (in terms of frequency, protocol, etc), zero otherwise. By abuse of notation, for each pass i we define $\mathcal{C}_i \subset A$ as the set of antennas, which are compatible with i. For our example scenarios, we assume that each pass is in a basic original schedule, assigned to an antenna, but that they are compatible with all the antennas.

With these data, the antenna-satellite scheduling problem can be informally stated as finding the times of connection of satellites (with visible and compatible antennas) verifying some performance criteria, which can include maximizing total connection time, enforcing fairness, trying not to deviate too much from previously used schedules, satellite preferences, and others, which are explained in subsequent sections. The main challenge arises from the fact that only one satellite can be connected at a time to a given antenna, and therefore many conflicts can emerge in scenarios with numerous satellites and scarce numbers of antennas.

B. Computation of Satellite Passes

To formalize the antenna-satellite scheduling problem, we first need to calculate the set of passes P that corresponds to the set of satellites N and (compatible) antennas A in the time horizon. A pass is defined as one window of visibility between a satellite and a given antenna at the ground station. Thus, given a time horizon, a satellite, and an antenna, several passes (or none) may be generated, depending on orbital mechanics and the underlying geometry.

The passes are indexed by i and the corresponding antenna k; the windows of visibility has an initial time value α_{ik}, and a final time value β_{ik}.

To compute these passes, we have to propagate the orbital elements of the satellites over the considered time horizon. If considering TLEs, one can use the simplified perturbations model SGP4 (see, e.g., [33]). The propagator calculates orbital state vectors of satellites relative to the Earth-centered inertial coordinate system (ECI) including the effect of the most relevant perturbations in LEO, and from there, computing the visibility is a simple geometrical calculation; for simplicity, a spherical Earth was considered in the process.

Pass computation was implemented as follows. Once we acquire the data of each satellite in TLE format, we use a two-step process. First, we propagate all the TLEs with a one minute sample time to delimit satellite visibility data. This is further refined, in the 1-min intervals where visibility is detected to be possible, with a much smaller sample time, obtaining sufficient accuracy with rapid computations time. When discussing the scenarios, the time spent obtaining the satellite visibility data is referred to as data acquisition time; this time is not taken into account in the total processing time. Fig. 1 shows the data acquisition time (on an average modern laptop, as described in Section IV) for a scheduling horizon of either one and four days (used in our example scenarios). Evidently, more exact propagators could be used at the expense of increasing these times, if precise initial orbital data are available.

C. Input Data Associated With Passes

Let N be a set of satellites and A the set of antennas, indexed by letters j and k, respectively. Following Section II-B, set of passes P is generated, which are indexed by i. The following data are associated with each pass:

1) As explained in Section II-B, $(\alpha_{ik}, \beta_{ik})$, with $\alpha_{ik} < \beta_{ik}$, is the time window on which the satellite generating pass i is visible by antenna k.
2) b_{ik} is the minimum time (in seconds) that antenna k needs to be connected to the satellite generating pass i to be able to download data. Evidently, the longer the connection time, the larger the amount of data downloaded.
3) $p_{ik} \geq 0$ is the priority of the pass for antenna k. Having $p_{ik} < p_{i'k}$ means that pass i is more preferred than pass i' for antenna k.
4) $d_{ik} = 1$ if the pass is preassigned to antenna k, and 0.5 otherwise. This parameter allows us to enforce keeping passes in their preassigned antennas, if possible, as in [30].

D. Computation of Conflict Pairs

After obtaining the passes, the algorithm needs as input the list of overlaps for a given antenna k; that is to say, how many potential "conflict pairs" exist, therefore creating a "conflict pair set," which we denote as $\mathcal{CF}_k$. This set contains all the pairs of passes i, i' that can overlap for a

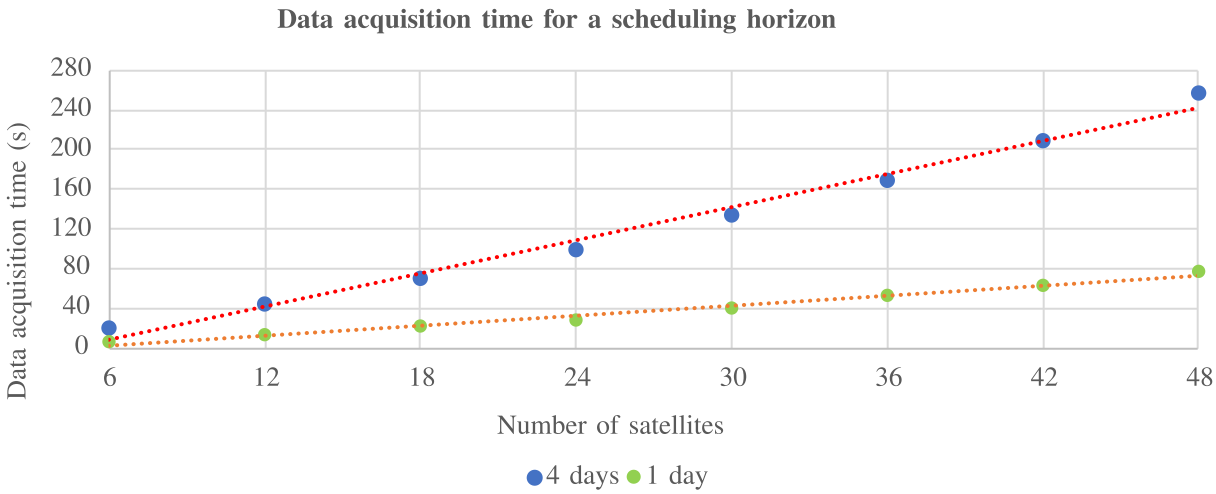


Fig. 1. Data acquisition time (on an average modern laptop, as described in Section IV) for a scheduling horizon of one day (green) and four days (blue). The dotted lines are linear fits to the data.

given antenna k. For this purpose, a simple function has been created which exploits the linear nature of the *timeline* of passes for each antenna instead of checking pairs of passes, which would slow down tremendously the processing time if many passes exists.

This process to compute potential conflict pairs is as follows. We start by storing for each antenna k, from zero and in ascending order, all the values of $(\alpha_{ik}, \beta_{ik})$ of the passes, creating an ordered vector of times with joint starting and ending times; the indexes of the passes are also put in the same order. All passes start as inactive. The timeline is then swept, starting at its initial value, until its end; if for a given pass, the starting time is the current value, then it is marked as active, but if the ending time is the current one, it is then marked as inactive. Every time a new pass is marked as active, one stores (as potential conflicts) all the pairs composed by this new active pass and all other passes, which are marked as active at the time.

We can see an example timeline with three passes and two antennas in Fig. 2. For simplicity, all passes can be on any of the two antennas. For both antennas, the computation is thus the same, and proceeds as follows. First, the time vector would be [0,200,300,400,500,600] and the associated index vector [3,1,2,3,1,2]. Starting the timeline, only Pass 3 is active. Then Pass 1 becomes active and one stores the pair (1,3). Next Pass 2 becomes active and one stores the pairs (1,2) and (2,3). No new activations happen, and all satellites become subsequently inactive, thus obtaining the list of potential conflicts as {(1,3),(1,2),(2,3)} (for both antennas).

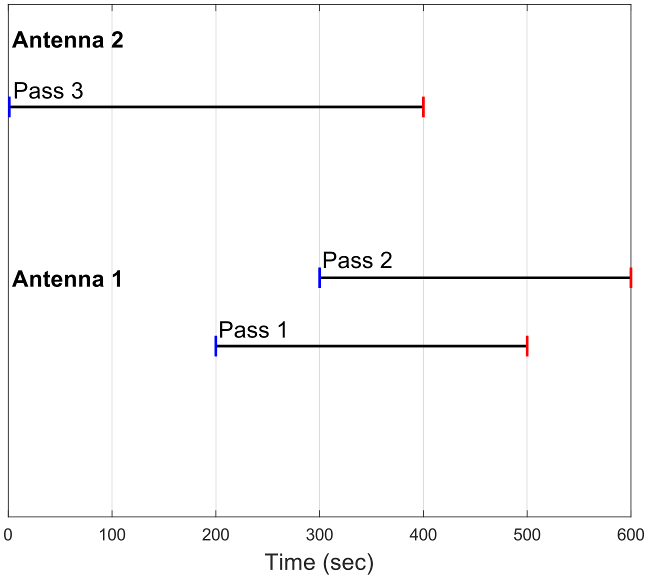


Fig. 2. Initial passes for Section II-D example.

III. SHAVING MODEL

In this section, we now present a new MILP model to solve the antenna-satellite scheduling problem, which we refer to as the shaving model. This new model has a main conceptual change with respect to the shortening model presented in [30]: Continuous variables to decide the starting time and end time of a pass-antenna connection are used. Thus, we avoid having the need to compute all possible intersections between passes, as required in [30] to formulate the ILP, and later in [32] which was our first attempt at a MILP model. This computation was extremely time-consuming when many passes are present. By avoiding it, the shaving model requires much less preprocessing time. The only time-consuming operation is the computation of conflict pairs, but as explained in Section II-D, we devised an efficient algorithm to reduce this burden.

In this section, we detail the variables, constraints, and objective functions used in our model to formalize the antenna-satellite scheduling problem.

A. Formulation

The model presented uses the following sets of variables:

1) $S_i \geq 0$ (resp. $E_i \geq 0$) defines the starting (resp. end) time of the connection between the satellite at pass i and the antenna assigned to it.
2) $V_{ik} = 1$ if pass i is assigned to antenna k, zero otherwise.
3) $W_{ii'} = 1$ if pass i is previous (or "to the left" if represented in a timeline, which is the nomenclature used in the sequel) to pass i' (this means that both passes are assigned to the same antenna, and the satellite generating pass i ends its connection before the one generating pass i' starts its, or in other words, the passes are not in conflict).

B. Objective Function

We consider a combination of different performance indexes to construct an objective function:

1) Moving performance: Following [30], we define

$$Z_1 = \sum_i \sum_{k \in C_i} (p^* - p_{ik} + 1) d_{ik} V_{ik}$$

where $p^* = \max_{ik} p_{ik}$. This function aims at maximizing the sum of priorities of satellite-antenna assignments while respecting as much as possible a prearranged schedule given by d_{ik}.
2) Shaving performance: $Z_2 = \sum_i (E_i - S_i)$. The aim of this function is to maximize the time that satellites are connected to antennas by keeping as much as possible of the original length of the passes.

The objective function of the shaving model is a weighted sum of both functions

$$\max (1 - \gamma) Z_1 + \gamma Z_2 \tag{1}$$

with $\gamma \in [0, 1]$. Note that if a pass i is cancelled (which means that it is not assigned to any antenna), then this pass contributes nothing to the objective function (because $V_{ik} = 0 \;\forall\; k \in \mathcal{C}_i$, and $E_i = S_i$). The assigned passes (which have not been cancelled), contribute in different degrees to the objective function, depending on whether they are assigned to one antenna or another, and their possible length.

C. Constraints

The set of constraints of the shaving model is as follows:

$$\sum_{k \in \mathcal{C}_i} V_{ik} \leq 1 \;\forall\; i. \tag{2}$$

$$E_i - S_i \geq \sum_{k \in \mathcal{C}_i} b_{ik} V_{ik} \;\forall\; i \tag{3}$$

$$E_i \leq S_{i'} + M(1 - W_{ii'}) \;\forall\; i \neq i' \tag{4}$$

$$W_{ii'} + W_{i'i} \geq (V_{ik} + V_{i'k}) - 1 \;\forall k \quad \forall\; i, i' \in \mathcal{CF}_k \tag{5}$$

$$S_i \geq \alpha_{ik} V_{ik} \qquad \forall\; i,\; k \in \mathcal{C}_i \tag{6}$$

$$E_i \leq \beta_{ik} + M(1 - V_{ik}) \qquad \forall\; i,\; k \in \mathcal{C}_i \tag{7}$$

$$E_i \leq \max_{k \in C_i} \beta_{ik} \sum_{k \in C_i} V_{ik} \quad \forall\; i. \tag{8}$$

Equation (2) ensures that each pass is assigned at most to one of its compatible antennas. Equation (3) imposes the minimum time that the satellite originating pass i must be connected to the antenna k to be considered a valid connection. Equation (4) enforces that, if $W_{ii'} = 1$ (i is to the left of i'), then pass i must end before the beginning of pass i'. M is defined as a sufficiently large positive number; in fact, $M = \max_k \beta_{ik}$ suffices. Equation (5) imposes that, if the passes i and i' belong to the conflict pair set for antenna k, then either i is to the left of i', or i' is to the left of i (note that using the conflict pair set avoids redundant constraints). Equations (6) and (7) ensure that, if the pass i is assigned to the antenna k, then it cannot start before the instant α_{ik} nor end after β_{ik}. In this case, $M = \max_k \beta_{ik}$ is also sufficient. Finally, constraints (8) make sure that $E_i = 0$ if pass i is not assigned to any antenna.

In addition, the following cut can be added, in order to accelerate the solution process without eliminating any optimal solution:

$$S_i \leq \max_{k \in C_i} \beta_{ik} \sum_{k \in C_i} V_{ik} \;\forall\; i. \tag{9}$$

Equation (9) causes that, if the pass i is not assigned to any antenna, then $S_i = 0$ (because S_i is defined as a nonnegative variable).

D. Comparison With Shortening Model

In this section, we briefly emphasize the main difference between the shaving model with respect to the shortening model. In the shortening model we need to compute *the subpasses*, which are defined as all possible intersections between distinct passes, as explained in [30]. Next, one has to assign one binary variable for each subpass-antenna combination. For example, for the passes in Example 1 (extracted from [30]) of Fig. 3 (left) one would need to create several subpasses for pass 2 as in Fig. 3 (right). Note that the computation of the set of subpasses is a challenging computational problem, since the number of subpasses originating from the (possibly large) set of passes may grow quite rapidly. Thus, the preprocessing step for the shortening model is already a computational burden (it may exceed the maximum allowed running time as shown in Section IV-B). Next, it is required to create a binary variable $Y_{ik} = 1$ if subpass i is assigned to antenna k, and zero otherwise. In the shaving model the subpasses are not required; only the much smaller set of passes is used. That makes the shaving model more efficient than the shortening model, as empirically proven in Section IV-B.

Additionally, there are some cases which cannot be solved by shortening without cancellation, as shown in Example 2 extracted from [32]. In this example, shown in Fig. 4, the candidate passes considerably overlap. Thus, shortening, shown in Fig. 5 can only keep one pass and has to cancel the other, whereas shaving, shown in Fig. 6, can adjust the durations and keep both passes.

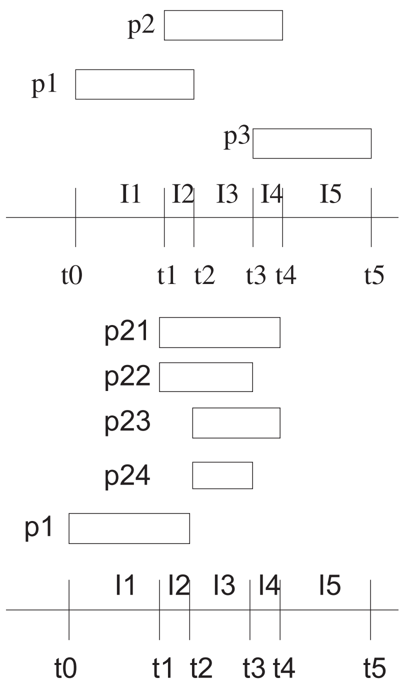


Fig. 3. Example 1 of Section III-D. Conflicts (top) and subpasses generated by pass 2 (bottom).

IV. Computational Experiments and Analysis

In this section we detail the experiments we have carried out in order to assess the quality of the model proposed in this paper, and compare it with that in [30]. All models and functions were coded in MATLAB R2021b using Gurobi 9.1.2, on a PC with Intel Core i7-11800H CPU (2.30 GHz).

A. Scenario Generation

A number of different scenarios with the following sizes have been generated:

1) Number of antennas: $|A| \in \{2, 4\}$, where $|A|$ denotes the cardinality of set A. The set of antennas A is composed of either two or four antennas, all located at the same site, Svalbard Satellite Station (SvalSat), with visibility in all directions; its closeness to the North Pole helps maximizing communications with EOSs, typically in Sun-synchronous (i.e., almost polar) orbits.
2) Number of satellites: $|N| \in \{6, 12, 18, 24, 30, 36, 42, 48\}$. The satellites used are EOSs, this is, LEO satellites mostly in sun-synchronous orbits. Their two-line element sets (TLE) are obtained from CelesTrak. We consider 48 satellites operated by different entities worldwide: Cosmo-SkyMed (from 1 to 4), Alsat-2 A, Cartosat (from 2 A to 2E), Deimos-1, U.K.-DMC 2, DubaiSat-1, ERS-2, Flock-3 M (from 1 to 4), H-2 A R/B, HaiYang-2 A, NigeriaSat-2, PRISM, RASAT, Resourcesat-2, Sentinel (1 A, 1B, 2 A, 2B, 3 A, 3B, 5P, and 6), SkySat (A, B and from C1 to C12), Sumbandila, TanDEM-X, Theos, and Tianhui-1. Note that some of those are already nonoperational, but we include them just for the purpose of experimentation.
3) Planning horizon (number of days): In our example scenarios, the interval's length is set to either one day or four days, this is, $D \in \{1, 4\}$, starting on the first of January of 2021 at 18:00 PM. Since we are considering LEO satellites in sun-synchronous orbits and a site close to the Earth's poles, we obtain about 14 passes each day per satellite.

In total, by combining the number of antennas (two possibilities), number of satellites (eight possibilities), and number of days (two possibilities), we generate $2 \times 8 \times 2 = 32$ different scenarios.

The chosen priorities p_{ik} depend on whether the pass' subindex is odd or even, that is, $p_{ik} = 5$ for even i, and $p_{ik} = 4$ for odd i.

Parameter d_{ik} is defined depending on the number of antennas. If we have two (resp. four antennas), the first two (resp. four) passes are preassigned to the first antenna, and the second two (resp. four) passes to the next one, and so on and so forth.

After the input data for the problem have been generated, we follow the steps of Section II to find the set of passes and the conflict pairs set.

B. Computational Results

Table II summarizes the results obtained in the 32 tested scenarios, with the last row showing average results over the first 30 scenarios, since the last two could not be solved by the shortening model. Column "Scenario" identifies the tested scenario as a three-tuple of the form $D - |N| - |A|$ (planning horizon, number of satellites, number of antennas). Column "m" refers to the resulting number of passes, which is an indicator of the problem size and complexity. Then, for each scenario and each model (shaving model or shortening model), in their corresponding row we use the following notation:

1) Column "ST" (shortened/shaved time) shows the total time (in hours) that visible satellites (i.e., satellites generating a pass) are not connected to any antenna. Evidently, the smaller this number, the better, as one wants to maximize the satellite connection times. Note that this indicator is related to objective Z_2.
2) Column "CP" shows the number of passes that have been cancelled. Since cancellations are undesirable, it becomes important to reduce this number as much as possible. Note that this indicator is related to objective Z_1.

Note that we imposed a maximum running time equal to 3600 s, which includes both preprocessing and the solver time; in the shaving algorithm, preprocessing is quite fast due to efficient computation of conflict pairs (see Section II-D) and, therefore most of the time is saved for the solver, while this may not be the case for the shortening model. Note also that, if the solver reaches the maximum

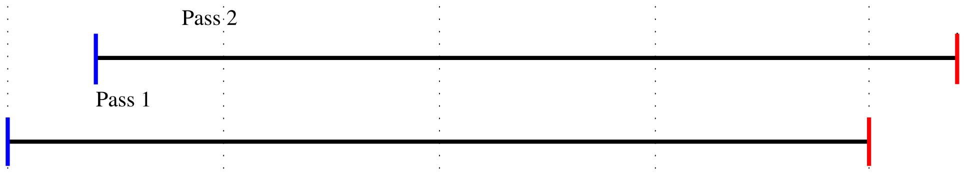


Fig. 4. Example 2 of Section III-D: Candidate passes with considerable overlap.

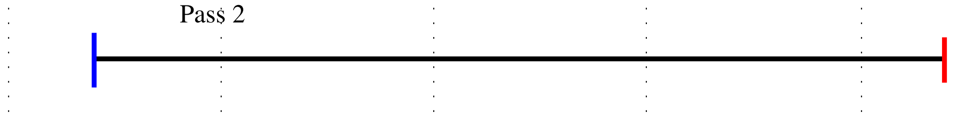


Fig. 5. Example 2 of Section III-D: Shortening solution cancels one pass.

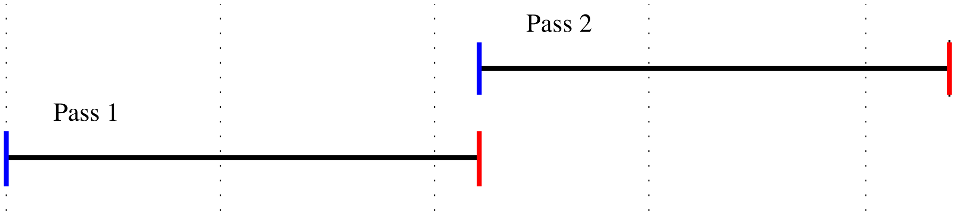


Fig. 6. Example 2 of Section III-D: Shaving solution adjusts the duration of the passes requiring no cancelations.

TABLE II
Complete Results Over the 32 Tested Scenarios for $\gamma = 0.5$

Instance		Shaving Model		Shortening Model	
Scenario	**m**	**ST**	**CP**	**ST**	**CP**
1-06-2	85	0.50	1	0.50	1
1-12-2	173	1.45	4	1.66	2
1-18-2	253	5.95	3	6.31	9
1-24-2	336	12.63	9	13.11	28
1-30-2	421	20.73	9	21.28	61
1-36-2	502	32.73	11	33.22	92
1-42-2	582	41.90	19	42.35	84
1-48-2	660	50.75	23	51.34	124
1-06-4	85	0.00	0	0.00	0
1-12-4	173	0.00	0	0.11	2
1-18-4	253	0.42	1	0.73	5
1-24-4	336	2.39	1	2.74	9
1-30-4	421	4.95	4	5.59	21
1-36-4	502	9.79	1	10.34	18
1-42-4	582	14.42	2	15.20	25
1-48-4	660	20.24	2	20.95	36
4-06-2	337	1.04	1	1.25	5
4-12-2	685	9.31	7	10.12	12
4-18-2	1004	26.21	19	27.56	37
4-24-2	1336	56.87	21	58.59	132
4-30-2	1660	85.67	30	87.22	226
4-36-2	1982	134.39	35	135.88	370
4-42-2	2300	171.26	75	172.86	525
4-48-2	2604	206.15	102	208.04	704
4-06-4	337	0.00	0	0.35	4
4-12-4	685	0.10	0	0.43	6
4-18-4	1004	1.31	1	2.46	20
4-24-4	1336	9.68	1	11.79	47
4-30-4	1660	18.50	4	20.76	55
4-36-4	1982	39.51	5	41.62	70
4-42-4	2300	57.34	7	-	-
4-48-4	2604	80.20	8	-	-
Average		32.63	13.03	33.48	93.3

running time, it returns a solution, which does not necessarily needs to be optimal. The last two scenarios could not be solved by the shortening model (due to the considerable number of passes, the preprocessing, which requires self-intersection of all passes, could not be finished).

Comparing the two models, we highlight that, on average, the shaving model with respect to the shortening model has the following advantages:

1) It reduces less time with respect to the visibility time windows (32.63 h against 33.48).
2) It cancels fewer passes (13.03 against 93.3).

Looking at the particular results for each scenario, we can see a few instances where the shaving model cancels more passes ({1-12-2}) or where the presence of sufficient antennas (4) for a few satellites (6 or 12) needs only a few operations ({1-06-4}, {1-12-4}, {4-06-4},{4-12-4}). Note that these cases are not very challenging given the rather small number of passes per day. However, as soon as the number of satellites and antennas grow large and numerous passes are generated (e.g., {4-42-4} or {4-48-4}), we can see that the shaving model offers excellent performance.

Regarding cancelled passes, one can see that the improvement is dramatic. Since fewer passes are cancelled, and satellites are connected for longer times, we can confidently state that the shaving model yields better results than the shortening model, even if more passes are shaved. Besides, the shaving model escalates much better than the shortening model (as suggested by the fact that, when the scenarios get larger, the shortening model is unable to return a feasible solution). We do not have results for shortening model in the last two scenarios because MATLAB ran out

TABLE III
Satellites With at Least One Cancelled Pass for Each Scenario

Scenario	Shaving Model	Shortening Model
1-06-2	1	1
1-12-2	3	2
1-18-2	2	7
1-24-2	6	13
1-30-2	8	25
1-36-2	8	29
1-42-2	13	36
1-48-2	19	43
1-06-4	0	0
1-12-4	0	2
1-18-4	1	5
1-24-4	1	9
1-30-4	4	19
1-36-4	1	17
1-42-4	2	22
1-48-4	2	28
4-06-2	1	4
4-12-2	5	9
4-18-2	14	16
4-24-2	9	24
4-30-2	15	30
4-36-2	22	36
4-42-2	34	42
4-48-2	43	48
4-06-4	0	3
4-12-4	0	5
4-18-4	1	12
4-24-4	1	18
4-30-4	4	23
4-36-4	4	30
4-42-4	21	-
4-48-4	23	-
Average	7.47	18.63

of memory, whereas the shaving model always returned a solution.

In addition, comparing the number of shortened passes in the shortening model versus the shaved passes in the shaving model, one can notice that, in average, 55% more passes were affected by the shaving operation. This is an expected outcome, being an indicator of how our model works, since to avoid cancellations one needs to better adjust the schedule by shaving more passes. In other words, both the total shaved time (ST column) and cancelled passes (CP column) are reduced due to a better distribution of shaved passes, meaning that more passes are shaved but with less total shaved time (ST column).

When analyzing the results at the satellite level instead of from the point of view of passes, Table III shows how, for the shaving model, the number of satellites with one (or more) cancelled passes also diminishes; indeed, the shortening model has about 2.5 times more satellites with at least a cancelled pass. Moreover, another notorious improvement from the point of view of satellites is a *fairer* distribution of cancelled passes, in terms of the satellites generating the passes. With the previous shortening model, it was common to find numerous cancelled passes generated by a single satellite; in fact, in some instances, up to half of the passes generated by a single satellite were cancelled. On the other hand, cancelled passes are quite uniformly distributed between satellites generating them with the shaving model.

TABLE IV
Effect of the Cut (9)

Scenario	With (9)	Without (9)	Difference (%)
1-06-2	0.1	0.2	100%
1-12-2	0.6	0.6	0%
1-18-2	6	6.2	3%
1-06-4	0.1	0.1	0%
1-12-4	0.8	0.8	0%
1-18-4	3	3	0%
1-24-4	63	92	46%
4-06-2	2	2	0%
4-12-2	9	9.4	4%
4-18-2	50	58	16%
4-06-4	1	1.2	20%
4-12-4	7	7	0%
4-18-4	22	27	23%
4-24-4	394	449	14%
Average			16%

The accumulation of cancelled passes belonging just to a few satellites becomes more evident in scenarios with two antennas, and tends to be proportional to the number of satellites. To illustrate this trend, we have extracted the number of cancelled passes per satellite for a particular scenario with numerous satellites, two antennas, and a time-span of one day, that is, scenario 1-48-2. The results are shown in Fig. 7, where it can be seen how cancelled passes accumulate for just a few satellites for the shortening model, whereas the shaving model has a *fairer* distribution of cancellations.

C. Effect of the Cut

In Section III, we claimed that (9) was a feasible cut for our problem. We now show that this set of constraints, indeed, does improve the efficiency of the model. The results shown before for the shaving model do include (9). In Table IV, we show a comparison regarding CPU time (in seconds) between the model with these constraints and the model without these constraints, with $\gamma = 0.5$. For each of the scenarios where both models found the optimal solution in less than 3600 s, we show the CPU time of the model with the cut, and that of the model without the cut. The last column computes the percentage difference between these two. The reader may note that in eight of the 14 such scenarios, the model with the cut found the optimal solution in less time, with an average percentage difference of 16%.

D. Effect of γ

In order to further test the efficiency of the new model, we run a new set of experiments in which all scenarios are solved by the two models, varying $\gamma \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Due to length constraints, we only show average results for each value of γ in Table V. On that table we see that, the shaving model always cancels less passes than the shortening model, and only for $\gamma = 0$ the shortening model reduces less connection time than the

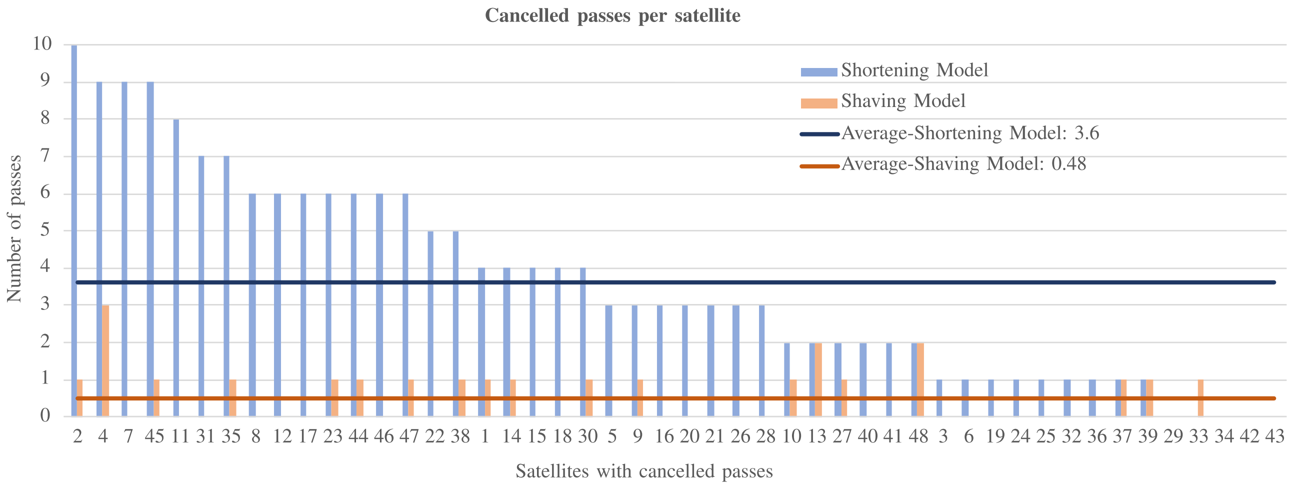


Fig. 7. Number of cancelled passes per satellite for scenario 1-48-2.

TABLE V
Effect of Parameter γ

	Shaving Model		Shortening Model	
γ	**ST**	**CP**	**ST**	**CP**
0	83.80	5.70	78.72	40.34
0.2	32.63	11.13	34.99	130.67
0.4	32.83	12.27	33.14	119.30
0.6	32.92	13.20	34.95	127.13
0.8	38.63	15.87	42.07	98.57
1	32.43	58.69	33.47	101.53
Average	42.21	19.48	42.89	102.92

shaving model. All in all, we can fairly state that the shaving model returns better solutions independently of the value of γ. It should also be emphasized that the shortening model is unable to return a feasible solution to the last two scenarios within one hour, regardless of the value of γ, whereas the shaving model finds feasible solutions to all scenarios and all values of γ.

V. CONCLUSION AND FUTURE WORK

In this article, we have introduced a new mixed integer linear programming model (named shaving model) to efficiently solve the antenna-satellite scheduling problem, which consists of scheduling abundant requests for allocation of antenna time slots by customers satellites. The problem complexity grows as the number of satellites increases. Our method has proven effective for 32 realistic scenarios, over an horizon of a few days. The results showed that the shaving model yields better solutions for larger scenarios, fairly lower shaved times, and much fewer cancelled passes than the previous shortening model, with much shorter preprocessing times and a *fairer* distribution of cancellations among satellites.

Future research will focus on the inclusion of additional criteria, such as, user technical specifications (e.g., limited storage), new variables that take into account unpredictable failures or changing the objective functions by only prioritizing the processes of shortening or moving. Another possible line could be the development of a metaheuristic algorithm, capable of rapidly producing feasible solutions for large scenarios.

REFERENCES

[1] A. J. Álvarez-Vázquez and R. S. Erwin, *An Introduction to Optimal Satellite Range Scheduling*. Berlin, Germany: Springer, 2015, vol. 106.

[2] L. Barbulescu, J.-P. Watson, L. Whitley, and A. Howe, “Scheduling space-ground communications for the air force satellite control network,” *J. Scheduling*, vol. 7, pp. 7–34, 2004.

[3] L. Barbulescu, A. E. Howe, J. P. Watson, and L. D. Whitley, “Satellite range scheduling: A comparison of genetic, heuristic and local search,” in *Proc. Int. Conf. Parallel Problem Solving Nature*, 2002, pp. 611–620.

[4] L. Barbulescu, A. Howe, and D. Whitley, “AFSCN scheduling: How the problem and solution have evolved,” *Math. Comput. Modelling*, vol. 43, pp. 1023–1037, 2006.

[5] L. Barbulescu, A. E. Howe, L. D. Whitley, and M. Roberts, “Understanding algorithm performance on an oversubscribed scheduling application,” *J. Artif. Intell. Res.*, vol. 27, pp. 577–615, 2006.

[6] B. Clement and M. D. Johnston, “The deep space network scheduling problem,” in *Proc. Innov. Appl. Artif. Intell.*, 2005, pp. 1514–1520.

[7] G. Corrao, R. Falone, E. Gambi, and S. Spinsante, “Ground station activity planning through a multi-algorithm optimisation approach,” in *Proc. IEEE First AESS Eur. Conf. Satell. Telecommun.*, 2012, pp. 1–6.

[8] S. Lee, W. C. Jung, and J.-H. Kim, “Task scheduling algorithm for the communication, ocean, and meteorological satellite,” *ETRI J.*, vol. 30, no. 1, pp. 1–12, 2008.

[9] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, “A Lagrangian Heuristic for satellite range scheduling with resource constraints,” *Comput. Oper. Res.*, vol. 38, pp. 1572–1583, 2011.

[10] F. Xhafa, X. Herrero, A. Barolli, and L. Barolli, “Evaluation of struggle strategy in genetic algorithms for ground stations scheduling problem,” *J. Comput. Syst. Sci.*, vol. 79, pp. 1086–1100, 2013.

[11] Z. Zhang, N. Zhang, and Z. Feng, “Multi-satellite control resource scheduling based on ant colony optimization,” *Expert Syst. Appl.*, vol. 41, pp. 2816–2823, 2014.

[12] N. Zufferey, P. Amstutz, and P. Giaccari, “Graph colouring approaches for a satellite range scheduling problem,” *J. Scheduling*, vol. 11, pp. 263–277, 2008.

[13] S. A. Chien et al., “A generalized timeline representation, services, and interface for automating space mission operations,” in *Proc. 12th Int. Conf. Space Oper.*, 2012, Art. no. 1275459.

[14] M. Schmidt, M. Rybysc, and K. Schilling, “A scheduling system for small ground station networks,” in *Proc. SpaceOps Conf.*, 2008, Art. no. 3578.

[15] M. Bester, “Automated multi-mission scheduling and control center operations at UC Berkeley,” in *Proc. IEEE Aerosp. Conf.*, 2009, pp. 1–12.

[16] M. Schmidt and K. Schilling, “A scheduling system with redundant scheduling capabilities,” in *Proc. IEEE Third Int. Conf. Space Mission Challenges Inf. Technol.*, 2009.

[17] X. Chen, G. Reinelt, G. Dai, and A. Spitz, “A mixed integer linear programming model for multi-satellite scheduling,” *Eur. J. Oper. Res.*, vol. 275, no. 2, pp. 694–707, 2019.

[18] P. Xie, H. Wang, Y. Chen, and P. Wang, “A Heuristic algorithm based on temporal conflict network for agile Earth observing satellite scheduling problem,” *IEEE Access*, vol. 7, pp. 61024–61033, 2019.

[19] X. Wang, G. Wu, L. Xing, and W. Pedrycz, “Agile Earth observation satellite scheduling over 20 years: Formulations, methods, and future directions,” *IEEE Syst. J.*, vol. 15, no. 3, pp. 3881–3892, Sep. 2021.

[20] B. Lemay, J. Castaing, R. A. Zidek, A. Cohn, and J. Cutler, “An optimization-based approach for small satellite download scheduling with real-world applications,” *J. Aerosp. Inf. Syst.*, vol. 15, no. 11, pp. 611–626, 2018.

[21] A. Sabol, R. Alimo, F. Kamangar, and R. Madani, “Deep space network scheduling via mixed-integer linear programming,” *IEEE Access*, vol. 9, pp. 39985–39994, 2021.

[22] T. Claudet, R. Alimo, E. Goh, M. D. Johnston, R. Madani, and B. Wilson, “△—MILP: Deep space network scheduling via mixed-integer linear programming,” *IEEE Access*, vol. 10, pp. 41330–41340, 2022.

[23] J. Kim, D.-H. Cho, J. Ahn, and H.-L. Choi, “Task scheduling of multiple agile satellites with transition time and stereo imaging constraints,” 2019, *arXiv:1912.00374*.

[24] M. Chen, J. Wen, B.-J. Pi, H. Wang, Y.-J. Song, and L.-N. Xing, “A novel genetic algorithm with population perturbation and elimination for multi-satellite TT&C scheduling problem,” in *Proc. Int. Conf. Bio-Inspired Comput., Theories Appl.*, 2019, pp. 558–568.

[25] W. Zhao, Y. Xue, G. Xia, S. Ren, and M. Zhang, “Resources scheduling algorithm for multi-relay-satellite system with multi-restriction,” in *Proc. Int. Conf. Intell. Transp., Big Data & Smart City*, 2019, pp. 451–455.

[26] H. Chen, Z. Luo, S. Peng, J. Wu, and J. Li, “HiPGen: An approach for fast generation of multi-satellite observation plans via a hierarchical multi-channel transformer network,” *Adv. Space Res.*, vol. 69, no. 8, pp. 3103–3116, 2022.

[27] J. Zhang, L. Xing, G. Peng, F. Yao, and C. Chen, “A large-scale multiobjective satellite data transmission scheduling algorithm based on SVM NSGA-II,” *Swarm Evol. Comput.*, vol. 50, 2019, Art. no. 100560.

[28] J. Zhang and L. Xing, “An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem,” *Comput. Oper. Res.*, vol. 139, 2022, Art. no. 105626.

[29] J. Wang, G. Song, Z. Liang, E. Demeulemeester, X. Hu, and J. Liu, “Unrelated parallel machine scheduling with multiple time windows: An application to Earth observation satellite scheduling,” *Comput. Oper. Res.*, vol. 149, 2023, Art. no. 106010.

[30] R. Vazquez, F. Perea, and J. Galan-Vioque, “Resolution of an antenna–satellite assignment problem by means of integer linear programming,” *Aerosp. Sci. Technol.*, vol. 39, pp. 567–574, 2014.

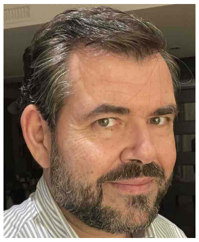
[31] F. Perea, R. Vazquez, and J. Galan-Vioque, “Swath-acquisition planning in multiple-satellite missions: An exact and heuristic approach,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1717–1725, Jul. 2015.

[32] J. I. Librero, R. Vazquez, and J. Galan-Vioque, “An algorithm for generation of antenna-satellite optimal schedules by using mixed integer linear programming,” in *Proc. Int. Workshops Plan. Scheduling Space*, 2015. [Online]. Available: https://sites.google.com/view/iwpss/proceedings/iwpss-2015-papers

[33] D. Vallado and W. McClain, *Fundamentals of Astrodynamics and Applications*, 3rd ed. Berlin, Germany: Springer, 2007.

Lorena Linares received the Bachelor degree in aerospace with the University of Sevilla, Sevilla, Spain, in 2020.

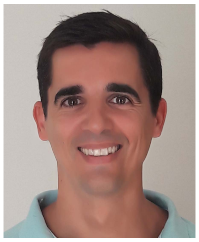
Her research interests include orbital mechanics, scheduling, optimization algorithms, and data analysis. She is currently working in industry in the area of Security and Export Control, Eurofighter Jagdflugzeug GmbH.



Rafael Vazquez (Senior Member, IEEE) received the B.Sc. degrees in electrical engineering and mathematics from the University of Seville, Sevilla, Spain, in 1998 and 2004, respectively, and the M.Sc. and Ph.D. degrees in aerospace engineering from the University of California, San Diego, La Jolla, CA, USA, in 2005 and 2006, respectively.

He is currently Professor with the Aerospace Engineering Department, University of Seville. His research interests include control theory, estimation, distributed parameter systems, and optimization, with applications to spacecraft and aircraft guidance, navigation and control, space surveillance and awareness, and scheduling for space missions.

He is coauthor of more than 150 journal and conference publications and the book Control of Turbulent and Magnetohydrodynamic Channel Flows (Birkhauser, 2007). He currently an Associate Editor for *Automatica* and *IEEE Control Systems Letters* .



Federico Perea received the bachelor’s degree in mathematics from the University of Sevilla, in 2001 and the Ph.D. degree in operations research from the Universidad de Sevilla, Sevilla, Spain, in 2007.

After being at Thales Nederland, The Netherlands, Universidad de Zaragoza, and Universitat Politècnica de Valencia, Spain, he currently is an Associate Professor with the Department of Applied Mathematics II, Universidad de Sevilla, and the General Secretary of the Institute of Mathematics, at the same university. He has published over 30 research papers in different areas of operations research.

Jorge Galán-Vioque received the bachelor’s and Ph.D. degrees in theoretical physics from the Autónoma University of Madrid, Madrid, Spain, in 1989 and 1993, respectively.

He is a Professor of mathematics with the Applied Mathematics Department, University of Sevilla, Sevilla, Spain. He was awarded a Fulbright Fellowship for a postdoctoral stay at Ohio State University. He has coauthored more than 60 research papers. His research interests include solid state theory, Hamiltonian dynamical systems with application to quantum and classical mechanics, and, recently, industry applied projects combining optimization, and orbital mechanics.