Aerospace Science and Technology 39 (2014) 567–574

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Resolution of an Antenna–Satellite assignment problem by means of Integer Linear Programming

Rafael Vazquez [a], Federico Perea [b], Jorge Galán Vioque [c,*]

[a] Departamento de Ingeniería Aeroespacial y Mecánica de Fluidos, Universidad de Sevilla, Spain
[b] Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, Spain
[c] Departamento de Matemática Aplicada II e Instituto de Matemáticas de la Universidad de Sevilla, Spain

A R T I C L E I N F O

Article history:
Received 20 February 2014
Received in revised form 20 May 2014
Accepted 20 June 2014
Available online 30 June 2014

Keywords:
Deconfliction
Integer programming
Antenna–Satellite allocation
Ground station operations
Automated scheduling

A B S T R A C T

Every day, ground stations need to manage numerous requests for allocation of antenna time slots by customers operating satellites. For multi-antenna, multi-site ground networks serving numerous satellite operators, oftentimes these requests yield conflicts, which arise when two or more satellites request overlapping time slots on the same antenna. Deconflicting is performed by moving passes to other antennas, shortening their duration, or canceling them, and has frequently been done manually. However, when many conflicts are present, deconflicting becomes a complex and time-consuming when done manually. We propose an automated tool that solves the problem by means of Integer Linear Programming. The models include operational constraints and mimic the manual process but consider the problem globally, thus being able to improve the quality of the solution. A simplified shortening model is also included to avoid excessive computation times, which is crucial given that the general problem has been reported NP-complete. Priorities are taken into account by tuning the cost function according to specifications of the requesting clients. Experiments with real-data scenarios using open-source software show that our tool is able to solve the Antenna–Satellite assignment problem for a large number of passes in a short amount of time, thus enormously improving manual scheduling operations, even when performed by a skilled operator.

1. Introduction

Satellite operators need the support of ground networks to perform key functions such as uploading commands or downloading gathered data. Recent years have seen a considerable growth in the number of satellites and their communication requirements, resulting in a substantial increase of requests for allocation of time slots in ground antennas. This increment of demand is even steeper for antennas located at strategic geographical locations—for instance, at sites nearby the poles, that provide multiple access windows per day to satellites in sun-synchronous orbits (which include the majority of Earth Observation Satellites, see [6]). At the same time that ground networks try to cope with demands by continuing to expand and build more sites throughout the world, the number of satellite customers keeps growing even faster. Also, new paradigms, such as distributed networks of small satellites [16], could push the networks' capabilities to the limit. Thus, ground station companies are faced with rather complex antenna–satellite allocation problems, not only due to a large number of requests compared with the limited number of resources, but also due to additional constraints originating from additional customers' requisites. Thus, the assignment procedure has become a rather cumbersome and time-consuming task when done manually, as it has often been resolved in the past.

The satellite–antenna assignment problem is often called the "Satellite Range Scheduling" (SRS) problem, and some resolving strategies have already been proposed in the literature. Barbulescu and coauthors have published many pioneering results in the area. For instance, Barbulescu et al. [2] solve the SRS to the US Air Force Satellite Control Network (AFSCN) in a scenario containing 100 satellites, 16 antennas, 9 stations, and 500 requests per day. With the objective of reducing the number of conflicts (typically 120), the authors find that genetic algorithms performed better than other alternatives. Subsequently, Barbulescu et al. [3] analyze the SRS both empirically and formally, proving that the problem is NP-complete, and provide new algorithms improving their previous results. Later, Barbulescu et al. [4] present the evolution of

* Corresponding author.
E-mail addresses: rvazquez1@us.es (R. Vazquez), perea@eio.upv.es (F. Perea), jgv@us.es (J. Galán Vioque).

http://dx.doi.org/10.1016/j.ast.2014.06.002
1270-9638/

the problem during 10 years at the AFSCN. They also analyze possible alternatives to the cost function, such as minimizing the sum of overlaps. The same group of authors study other heuristics for the SRS in Barbulescu et al. [5], by combining several algorithms.

A number of published works by other authors also deserve mention. For instance, Clement and Johnston [9] describe the SRS for the Deep Space Network (DSN) considering a scenario with 16 antennas, 20 spacecrafts, four-month time-frames, and 650 passes per week. They generate and repair schedules, and pose heuristics for solving the problem with emphasis in re-scheduling. Corrao et al. [10] integrate Genetic Algorithms, Graph Theory and Linear Programming in order to build conflict-free plans, and apply their approach to a practical case study provided by a satellite service company. Lee et al. [14] study the scheduling of a single geostationary satellite. Marinelli et al. [15] formulate the problem as an ILP model, which is found infeasible and then solved by means of a Lagrangian relaxation. As a case study, they apply their approach to Galileo. Xhafa et al. [21] solve SRS by using Struggle Genetic Algorithms on STK simulations. Zhang et al. [22] propose ant-colony algorithms, solving examples with 17 satellites and 11 to 13 antennas, yielding around 400 passes. Zufferey et al. [23] apply graph coloring algorithms to a set of 500 realistic instances. Finally, Chien et al. [8] take a more global point of view and 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 [18,7] for small satellites operated by research institutions, which usually have some specific needs such as redundancy and flexibility. Schmidt and Schilling[17] solve this problem with a tailored approach that also maximizes redundancy in order to solve possible failures in communication, and consider a simple scenario with 6 satellites and 4 stations, yielding 51 contact windows.

However, due in part to tradition, and in part to the complexities of the problem, manual handling of schedules is still routine for ground networks managers. To simplify the procedure, they plan a batch of antenna–satellite assignments by starting from the last available schedule. Recomputing the satellite positions from their orbits gives the observation windows (these are time intervals of accessibility computed from the satellite orbit, the antenna geographical location, and allowable positions of Azimuth-Elevation for the antenna, i.e., which region of the sky is accessible for the antenna). We refer to these windows as “a pass”. Since passes usually differ from those obtained in past schedules (due to movement along the orbit and the rotation of the Earth), a previous schedule is normally not reusable in the future. Each pass has a default antenna, which is the one requested by the user; this would be considered as the most preferred antenna. From the passes and the users' preferences, antennas are initially assigned. Since the passes for different satellites can partly coincide in time, and different users often select the same preferred antenna, initial allocations may cause *conflicts*, i.e., time intervals where different passes overlap on the same antenna. Such conflicts can be addressed by performing what we refer to as “deconflicting,” which can be carried out by using certain operations on the passes (which we call deconflicting operations). First, the most preferred option would be just reallocating some passes to other compatible[1] antennas located at the same site. Other options in order of preference would be moving the pass to another site (which could however imply considerable changes in the time allocation if the new site is far away), shortening the pass (up to a minimum duration, as requested by the user), or, if no other options are available, canceling the pass. Some of the deconflicting operations might be performed only on a subset of passes if there is a number of already allocated passes that must be honored (for instance for preferred clients or previous commitments); we denote those as “accepted” passes.

Network operators perform manual deconfliction by reviewing conflict after conflict, in an order that takes into account that some satellites (or customers) have a higher priority than others. To solve the conflicts, they perform the deconflicting operations that are allowed for the involved passes, in the preferred order. However, since they are sequentially processing the conflicts and not considering the problem in a global fashion, they often end up canceling passes that could be otherwise accommodated by using a more systematic procedure able to maximize some measure of performance.

A similar problem to ours is the disjunctive scheduling problem (DSP). The SRS we study in this paper and the DSP share that a set of tasks (in our case passes) have to be assigned to a machine (in our case antennas). In DSP tasks cannot be interrupted, just like the connection between passes and antennas. They also share the “disjunctive” feature, that is, two different tasks (passes) cannot be processed at the same time in the same machine (antenna). On the other hand, there are some discrepancies. First of all, the traditional objective in DSP (see [12]) is the minimization of the *makespan*, the completion time of the latest task, which is different from the objectives considered in this paper. Secondly, unlike the DSP, our SRS does not impose precedence constraints. The interested reader is referred to [1] and [19] for more insights into the DSP and algorithms for solving it.

In this paper we propose a procedure to solve a problem of deconfliction that was posed by a ground station operator managing an extensive network, composed by several sites with dozens of antennas, from now on called “the company”. Even though the general scheduling problem has been reported NP complete (see [3]), we have found success in solving the problem by using exact Integer Linear Programming (ILP) models. This is due to the fact that we base our models in the formulation used by the company, which is more specific and restrictive than the general SRS formulation, in the sense that many passes do not have more than one or two antenna alternatives, and user preferences strongly shape the resulting solution. In addition, we model the shortening deconflicting operation in a simplified way that avoids the use of continuous variables. Using our models, we have been able to solve in a reasonable time (less than a minute) real-world instances of the problem over a time frame of about a week, by using an open source ILP solver. The instances were of considerable dimensions (thousands of passes over dozens of antennas), with hundreds of conflicts, and provided by the company; their manual resolution by a skilled operator took about one entire day of work. The use of ILP models has proven fruitful for other space mission optimization problems, such as the problem of swath acquisition planning for multiple Earth Observation Satellites, see for instance [13].

The remainder of this paper is structured as follows. In Section 2, the problem is formally stated and the notation used throughout the paper is introduced. The different deconfliction objectives, and the resulting models are described in Section 3, formulated as Integer Linear Programming problems. Computational results, taken from real data, are analyzed in Section 4. We finish with some concluding remarks in Section 5.

2. Problem setting

In this section we formulate the Antenna–Satellite assignment problem. We begin by listing the basic input data required from satellites and antennas. Then, we explain how to compute the

[1] A given satellite communication requirements can often be supported just by a subset of the available antennas in a site.

passes and calculate possible conflicts by using time intervals of the passes.

2.1. Input data

The input data of the problem are:

1. The time-frame for the planning problem, which is an interval $[T_0, T_f]$ given, respectively, by the initial and final times T_0 and T_f. We refer to this interval as T (in our case, usually a week).
2. A set $\mathcal{S}$ of satellites. The orbits of the satellites can be given in any conventional format, for instance as Two-Line Elements (TLEs) on a certain epoch (which should be close to T to be able to precisely determine the passes).
3. A set $\mathcal{A} = \{A_1, \ldots, A_{n_a}\}$ of antennas, given by their geographical locations. Antennas which are geographically close to each other are considered to be in the same site, whereas antennas located far away from each other are in different sites. For each antenna, we also assume that we know its admissible range of Azimuth-Elevation, which would model obstacles and local geography (for instance mountains), and the required minimum elevation above the horizon to avoid atmospheric effects. This is mathematically formulated as the set $\Omega_a = \{(\mathrm{Az}, \mathrm{El})\}$ of accessible points in the sky given by their azimuths and elevations.
4. The set of compatible antenna–satellite pairs $\mathcal{C} \subset \mathcal{A} \times \mathcal{S}$.

2.2. Computation of passes

The next step to formulate the problem is to calculate the set of possible passes for all satellites $\mathcal{S}$ over the antennas $\mathcal{A}$ in the given time-frame. For each revolution of a satellite $s \in \mathcal{S}$ over the Earth, we obtain a pass P when there are time intervals of the form $[t_0, t_1] \subset T$ during which a satellite is accessible for one or more antennas $a \in \mathcal{A}$, given that the duration of the accesses, $t_1 - t_0$, is greater or equal than the minimum duration $t_{\min_{sa}}$ and the antenna is compatible with the satellite requirements, i.e. $(a, s) \in \mathcal{C}$. We assume that there is an antenna to which the pass is originally assigned; other possible antennas to which the pass can be assigned are called alternative antennas.

To perform this computation, the first step is to propagate the orbital elements of the satellites during the mission time-frame. This can be done using any of the many possible methods available in the literature, which incorporate more or less accurate models of orbit perturbations (see for instance [20], and references therein). Once the elements are known at all times $t \in T$, the vector position $\vec{r}_s(t)$ in the geographical reference frame (that rotates with the Earth) can be computed [11], for all $s \in \mathcal{S}$. Then, using the antenna geographical coordinates the vector position of the antennas $\vec{r}_a$ for all $a \in \mathcal{A}$ can be also computed. Then, by projecting the relative position of the satellite with respect to the antenna, $\vec{r}_{as}(t) = \vec{r}_s(t) - \vec{r}_a$ on the topocentric frame centered in the respective antenna, one can compute the azimuth and elevation for each compatible antenna–satellite pair, $(\mathrm{Az}_{as}(t), h_{as}(t))$ for $(a, s) \in \mathcal{C}$. Each of the time intervals in which $(\mathrm{Az}_{as}(t), h_{as}(t)) \in \Omega_a$ for at least the minimum duration $t_{\min_{sa}}$ constitutes a pass. Satellites generate a pass only each time the groundtrack approaches a given antenna. For most locations, this would happen at most once or twice a day. However, oftentimes satellites have sun-synchronous orbits, the most frequently used orbit for Earth Observation Satellites (due to constant lighting properties). These satellites transmit large amounts of data and therefore constitute a large subset of the satellites requesting antenna time slots. Given that sun-synchronous satellites have almost-polar low orbits, sites close to the poles of the Earth would obtain passes on most orbit revolutions (around 13 passes each day).

It is important to note that this computation is time-frame dependent. Given that the satellites are in different orbits, the ordering and length of the passes will be different each time-frame. Therefore for different time-frames the problem will present changes that, depending on the relative orbits, might totally modify the inputs of the problem. Thus, the solution for one time-frame is not applicable, in general, to another.

2.3. Additional input data for the passes

Once the passes have been computed, we obtain a set $\mathcal{P} = \{P_1, \ldots, P_{n_p}\}$ consisting of n_p passes. The time interval during which each of these passes P_i can access a given antenna A_k is given by the intervals $[\alpha_{ik}, \beta_{ik}]$. Passes are classified as accepted or free. In the former, the requested antenna and time slot are considered to be fixed, while in the latter, one is allowed to change antenna, to shorten the duration of the pass, or even to cancel it, in order to deconflict. The following parameters are additional input data for our problem:

1. $F \subset \{1, \ldots, n_p\}$ is the set of free passes, i.e., passes which can be modified with respect to the original request. Note that this set does not include all passes in $\mathcal{P}$, as some of them are assigned to antennas by the company and such assignment cannot be changed.
2. p_{ik}: priority of pass P_i in antenna A_k. $p_{ik} < p_{i'k}$ means that P_i is more preferred than $P_{i'}$ for antenna A_k.
3. a_{ik}: minimum length of time in which P_i must be active if antenna A_k is to get its data. Such length of time includes pre- and post-processing times required by tracking functions, which depend on the satellite–antenna pair.
4. The binary parameter e_{ik} takes the value 1 if pass P_i is originally requested to be assigned by default to antenna A_k. We assume that $\sum_{k=1}^{n_a} e_{ik} = 1$ for every pass P_i, that is, originally pass P_i is assigned to one and only one antenna.
5. C_i is the set of antennas which have access and are compatible with pass P_i. The binary parameter c_{ik} takes value 1 if pass P_i has access and is compatible with antenna A_k, and 0 otherwise. In other words, $c_{ik} = 1$ if and only if $k \in C_i$.
6. $(\alpha_{ik}, \beta_{ik})$ is the period of time in which pass P_i has access to antenna A_k, $k \in C_i$.

2.4. Computation of time intervals

Once all the passes have been computed and the input data on the passes has been gathered, the formulation of the Antenna–Satellite assignment problem requires finding the different time intervals during which, for a given antenna, the passes can overlap. We call this the “antenna timeline”.

Thus, for each antenna A_k, we consider the intersections of all possible intervals of time $(\alpha_{ik}, \beta_{ik})$ of compatible passes. The result is n_s ($n_s \le 2n_p - 1$) intervals $I_{1k}, \ldots, I_{n_s k}$, with lengths $l_{1k}, \ldots, l_{n_s k}$. The intervals are sorted in such a way that the beginning of interval I_{jk} is equal to the end of interval $I_{j-1,k}$, or larger if there is a “gap” during which no compatible passes exist for the antenna.

Then, $S_{ik} \subset \{1, 2, \ldots, n_s\}$ is the set of indices j of the sorted intervals $\{I_{jk},\ \ j = 1, \ldots, n_s\}$, in which P_i can be active in antenna A_k. This is computed by taking into account the accepted (fixed) passes; by construction, when an antenna is already accessing an accepted pass which overlaps with the interval I_{jk}, we have $j \notin S_{ik}$.

We show a simple example of such a timeline in Fig. 1, which considers three passes (p_1, p_2 and p_3) and two antennas (A_1 and

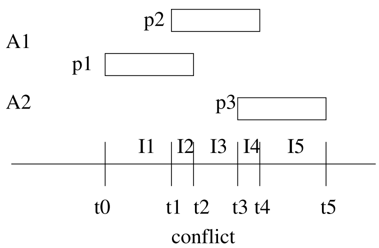

Fig. 1. Simple example of construction of time intervals and conflict.

A_2). This example is used later for demonstrating the ILP formulation. For the sake of simplicity the passes could be located at either antenna with the same start and end times. From the figure we see that the beginning of p_1 in either antenna is $\alpha_{11} = \alpha_{12} = t_0$, the ending of p_1 is $\beta_{11} = \beta_{12} = t_2$, and similarly, for p_2 we have $\alpha_{21} = \alpha_{22} = t_1$ and $\beta_{21} = \beta_{22} = t_4$, and for p_3 we have $\alpha_{31} = \alpha_{32} = t_3$ and $\beta_{21} = \beta_{22} = t_5$. The resulting intervals for both antennas are $I_{11} = I_{12} = [t_0, t_1]$, $I_{21} = I_{22} = [t_1, t_2]$, $I_{31} = I_{32} = [t_2, t_3]$, $I_{41} = I_{42} = [t_3, t_4]$ and $I_{51} = I_{52} = [t_4, t_5]$. Assuming no passes are fixed we have $S_{11} = S_{12} = \{1, 2\}$, $S_{21} = S_{22} = \{2, 3, 4\}$ and $S_{31} = S_{32} = \{4, 5\}$, which means that the first pass spans (in either antenna) the time intervals 1 and 2, the second pass the time intervals 2, 3 and 4, and third pass the time intervals 4 and 5. If Fig. 1 represents the originally proposed schedule then we see there is a conflict in antenna 1 between passes 1 and 2, which can be trivially resolved either by moving pass 1 to antenna 2 (solution 1) or by switching antenna between passes 2 and 3 (solution 2), as show in Fig. 2.

3. ILP models for the Antenna–Satellite assignment problem

A conflict is produced if there is an overlap, i.e., when during the same time interval, two passes are assigned to the same antenna. Using our notation, this occurs when there exists an antenna A_k and an interval I_{jk} such that

$$\sum_{i \in \mathcal{P}: j \in S_{ik}} e_{ik} > 1.$$

In words, when the sum of all e_{ik} for all passes P_i such that P_i can be active in interval I_{jk} is greater than one, for a given antenna A_k and a time interval I_{jk}, there is a conflict in this antenna in this time interval.

The object of this paper is to develop an algorithm that finds a feasible solution for the Antenna–Satellite assignment problem keeping as many conflict-free passes allocated as possible. In a sense, this is equivalent to minimizing the number of cancellations. However, there are some additional considerations: First, not all the passes have the same priority. Thus, the operator (user) may accept the cancellation of a higher number of passes provided that the more preferred (economically more valuable) passes' requests are satisfied. Additionally, there is a company-defined hierarchy of actions to deconflict the passes, which is introduced next.

When conflicts are found, one of the following three deconflicting operations is done:

1. Moving passes to a different antenna (see Section 3.1) at the same site or at another site.
2. Shortening the passes' time slot on the antenna (Section 3.2).
3. Cancellation of passes (Section 3.3).

These operations are listed in order of preference, i.e., first, if possible, conflicts should be addressed by moving passes to antennas different to the default ones (and if possible within the same site). Only if does not solve all conflicts, passes should be shortened (when admissible, and taking into account the minimum duration of a pass). Still, if this operation is not enough, some passes can be canceled to find a feasible conflict-free solution.

In what follows we formulate a global ILP model that simultaneously includes all these deconflicting operations. The possible solutions are weighted in the cost function to reflect the preference of the different operations, while at the same time taking into account the priority of the different satellites. The models are individually presented in the same order of operation preference for the sake of clarity, but it must be understood that the final model considers all three operations simultaneously, thus aiming to obtain the "best" possible global solution.

3.1. Moving passes to a different antenna

We address first the problem of permutating free passes between antennas so that conflicts disappear, while taking into account the priorities of the different passes.

Define, for each $i \in F$, and for each $k \in C_i$, (i.e., for each free pass and compatible antenna) the binary variable y_{ik} which takes the value 1 if pass P_i is assigned to antenna A_k and 0 otherwise.

The constraints of the model would be:

1. Every free pass has to be assigned to one and only one antenna.

$$\sum_{k \in C_i} y_{ik} = 1, \quad \forall i \in F. \tag{1}$$

2. For a given antenna A_k and a time interval I_{jk} available for free passes, i.e., with $\bigcup_{i \in F} S_{ik} \neq \emptyset$, there should be no conflict among the n_p passes.

$$\sum_{i \in F:\, j \in S_{ik}, k \in C_i} y_{ik} \leq 1, \quad \forall\, k, j : \bigcup_{i \in F} S_{ik} \neq \emptyset. \tag{2}$$

The sum in this equation is taken for all free passes $P_i \in F$ such that this assignment can be done during time interval I_{jk} and antenna A_k is compatible with pass P_i.

The objective is to keep as many passes allocated to the requested antennas as possible, taking into account the different priorities. We model this as the maximization of the cost index J_1

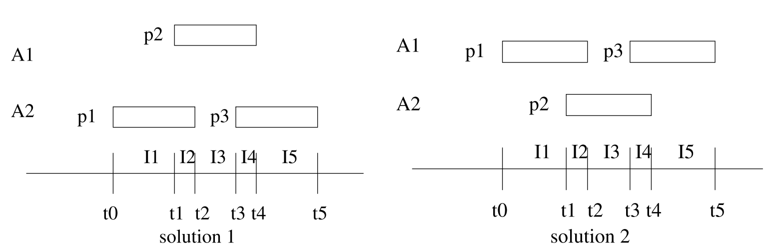

Fig. 2. Two possible solutions for the example.

defined as the sum of priorities of passes[2] that remain assigned to the requested antennas.

$$J_1 = \sum_{i\in F}\sum_{k\in C_i}(p^* - p_{ik} + 1)\xi_{ik} y_{ik}, \tag{3}$$

where $p^* = \max_i p_i$ and ξ_{ik} is a weighting function that is defined in the examples of Section 3 as

$$\xi_{ik} = \begin{cases} 1, & e_{ik} = 1, \\ 1/2, & e_{ik} = 0,\ A_k \text{ in the same site}, \\ 1/4, & e_{ik} = 0,\ A_k \text{ in a different site}. \end{cases} \tag{4}$$

Thus we favor to stay in the initially assigned antenna and penalize changing antenna and site, while at the same time enforcing priorities. In Section 4 we explain the values of ξ_{ik} that were used in the experiments. Note also that the priorities p_{ik} and the weights ξ_{ik} are not redundant as they do not have the same weight in the cost function. On one hand, the company would prefer that the solution remains as close to the initial assignment as possible, which is weighted equally for all passes (the ξ_{ik} coefficients). On the other hand, for economic reasons (some contracts being more expensive or strategically important than others), there is a hierarchy of passes based on priorities, which implies a very diverse set of weights (the p_{ik}) that has a considerable impact on which passes are canceled and which are not.

In the example presented in Fig. 1, we have $e_{11} = e_{21} = e_{32} = 1$, and $e_{12} = e_{22} = e_{31} = 0$. Assume for simplicity that all preferences are the same and equal to 1 and that both antennas are located in the same site. Then we have 6 binary variables y_{11}, y_{12}, y_{21}, y_{22}, y_{31}, y_{32}. Constraint (1), which implies that every pass is assigned only to one (compatible) antenna, reads

$$y_{11} + y_{12} = 1, \tag{5}$$

$$y_{21} + y_{22} = 1, \tag{6}$$

$$y_{31} + y_{32} = 1. \tag{7}$$

Constraint (2), which implies that a solution has no overlaps, is constructed by looking at the potential overlaps interval by interval, and would read

$$y_{11} + y_{21} \le 1, \tag{8}$$

$$y_{21} + y_{31} \le 1, \tag{9}$$

$$y_{12} + y_{22} \le 1, \tag{10}$$

$$y_{22} + y_{32} \le 1. \tag{11}$$

Finally the cost index (3) becomes

$$J_1 = y_{11} + \frac{y_{12}}{2} + y_{21} + \frac{y_{22}}{2} + \frac{y_{31}}{2} + y_{32}. \tag{12}$$

Thus, maximizing J subject to constraints (5)–(11) yields $y_{12} = y_{21} = y_{32} = 1$ and $y_{11} = y_{22} = y_{31} = 0$, i.e., the solution shown in Fig. 2 (left), which represents a conflict-free solution optimizing the number of changes according to (3).

3.2. Shortening the duration of passes

We now present a model that includes in addition the possibility of shortening.

The idea is to shorten a pass by reducing the set of time-intervals that it covers (but not creating new time-intervals). Formalizing this idea, for each pair of compatible pass-antenna, P_i, A_k,

[2] In fact, in J_1—which is to be maximized—the priorities p_{ik} are transformed to $p^* - p_{ik} + 1$, where p^* is the maximum value of priority, to reflect the fact that smaller values of p_{ik} represent a higher priority, i.e., a more desirable pass.

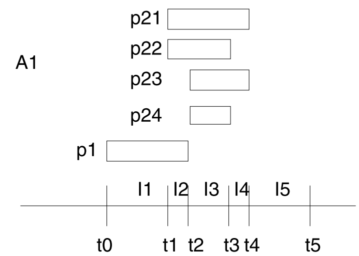

Fig. 3. Construction of subpasses (only shown for pass 2 in antenna 1).

and for each connected[3] subset $S^\ell_{ik} \subset S_{ik}$ that satisfies $\sum_{j\in S^\ell_{ik}} l_{jk} \ge a_{ik}$, we define a subpass P^ℓ_{ik} which spans a time interval $[\delta^\ell_{ik}, \rho^\ell_{ik}]$.

Now we have to choose one such subpass for each pass P_i so that the objective function is optimized, and no conflicts arise. For each subpass P^ℓ_{ik} we define the binary variable $y_{i\ell k} = 1$ if subpass P^ℓ_{ik} is selected, and zero otherwise.

The objective is now twofold: to maximize the active time of passes and to keep passes in the antennas they were originally assigned to or at least keep the preference as in Section 3.1. This is modeled by defining an additional cost index

$$J_2 = \sum_{i,\ell,k}(\rho^\ell_{ik} - \delta^\ell_{ik}) y_{i\ell k},$$

which counts the active time of passes, and later on maximize a linear combination of J_1 and J_2 defined as:

$$(1-\gamma)J_1 + \gamma J_2, \tag{13}$$

where parameter $\gamma \in [0, 1]$ measures the importance given to each of the two objectives ($\gamma = 0$ means that the only objective is to keep as may passes allocated to the requested antennas as possible, $\gamma = 1$ means that the only objective is to maximize the active time of passes, any other $\gamma \in (0, 1)$ maximizes a combination of both objectives). In Section 4 we give the value of γ that was used in the experiments.

The constraints of the model are the constraints of Section 3.1 substituting the constraints that include shortened passes by

$$\sum_{\ell,k} y_{i\ell k} = 1, \quad \forall i, \tag{14}$$

$$\sum_{i\in F: j\in S_{ik}, k\in C_i} y_{ik} + \sum_{i,\ell: j\in S^\ell_{ik}} y_{i\ell k} \le 1, \quad \forall k, j. \tag{15}$$

The meaning of (14) is that, for each pass, exactly one subpass is selected and assigned to exactly one antenna. On the other hand, with (15) we impose that, for each antenna and each interval, at most one pass or shortened subpass is active.

In the example presented in Fig. 1, we could force the necessity of shortening if we assume, for instance, that pass 1 is compatible only with antenna 1 and pass 3 is compatible only with antenna 2. Thus the variables y_{12} and y_{31} disappear and constraints (5)–(11) are reduced to

$$y_{11} = 1, \tag{16}$$

$$y_{21} + y_{22} = 1, \tag{17}$$

$$y_{32} = 1, \tag{18}$$

$$y_{11} + y_{21} \le 1, \tag{19}$$

$$y_{22} + y_{32} \le 1. \tag{20}$$

[3] A set of intervals is considered connected if their union forms a unique interval.

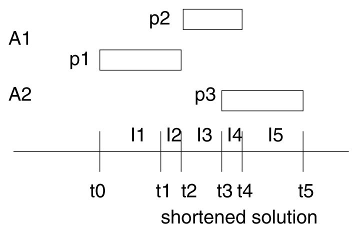

Fig. 4. Simple example of a shortened solution.

Notice that this results in an infeasible problem; it is not possible to use any of the solutions pictured in Fig. 2 as there is no antenna to which we can move pass 2 without overlapping with other passes. Thus we need to include the possibility of shortening. To simplify, assume also that passes 1 and 3 cannot be shortened. Thus one considers shortening pass 2. The possible subpasses would span the following time intervals: $[t_1,t_4]$, $[t_1,t_3]$, $[t_1,t_2]$, $[t_2,t_4]$, $[t_2,t_3]$, $[t_3,t_4]$. Assuming that, out of these, $[t_1,t_2]$ and $[t_3,t_4]$ are too short to be considered, we end up with four potential subpasses for each antenna, namely P^1_{21}, P^2_{21},P^3_{21},P^4_{21} and P^1_{22}, P^2_{22},P^3_{22},P^4_{22}, and the corresponding binary variables y_{211}, y_{221}, y_{231}, y_{241} and y_{212}, y_{222}, y_{232}, y_{242}. The possible subpasses are shown in Fig. 3 (only for antenna 1 to avoid cluttering the figure). The respective intervals would be $[\delta^1_{21},\rho^1_{21}]=[t_1,t_4]$, $[\delta^2_{21},\rho^2_{21}]=[t_1,t_3]$, $[\delta^3_{21},\rho^3_{21}]=[t_2,t_4]$, $[\delta^4_{21},\rho^4_{21}]=[t_2,t_3]$, and equally $[\delta^1_{22},\rho^1_{22}]=[t_1,t_4]$, $[\delta^2_{22},\rho^2_{22}]=[t_1,t_3]$, $[\delta^3_{22},\rho^3_{22}]=[t_2,t_4]$, $[\delta^4_{22},\rho^4_{22}]=[t_2,t_3]$. The new set of constraints would be

$$
\begin{aligned}
&y_{11}=1,\\
&y_{211}+y_{221}+y_{231}+y_{241}\\
&\quad+y_{212}+y_{222}+y_{232}+y_{242}=1,\\
&y_{32}=1,\\
&y_{11}+y_{211}+y_{221}\le 1,\\
&y_{211}+y_{221}+y_{231}+y_{241}\le 1,\\
&y_{211}+y_{231}+y_{241}\le 1,\\
&y_{211}+y_{231}\le 1,\\
&y_{212}+y_{222}\le 1,\\
&y_{212}+y_{222}+y_{232}+y_{242}\le 1,\\
&y_{212}+y_{232}+y_{242}\le 1,\\
&y_{32}+y_{212}+y_{232}\le 1.
\end{aligned}
$$

The cost function now becomes

$$
\begin{aligned}
J={}&y_{11}+y_{32}+\big(\gamma(t_4-t_1)+1-\gamma\big)y_{211}\\
&+\big(\gamma(t_3-t_1)+1-\gamma\big)y_{221}+\big(\gamma(t_4-t_2)+1-\gamma\big)y_{231}\\
&+\big(\gamma(t_3-t_2)+1-\gamma\big)y_{241}+\left(\gamma(t_4-t_1)+\frac{1-\gamma}{2}\right)y_{212}\\
&+\left(\gamma(t_3-t_1)+\frac{1-\gamma}{2}\right)y_{222}+\left(\gamma(t_4-t_2)+\frac{1-\gamma}{2}\right)y_{232}\\
&+\left(\gamma(t_3-t_2)+\frac{1-\gamma}{2}\right)y_{242}.
\end{aligned}
$$

The solution would depend on the particular values of γ and the times. If one chooses $\gamma=1/2$ and the time intervals are to scale in Fig. 1, then the solution is to choose subpass P_{231} as shown in Fig. 4.

3.3. Canceling passes

If it is required to resolve all conflicts, the pass(es) with lowest priority might be canceled. For this situation we use the same variables as in Section 3.1 and Section 3.2. To find a feasible solution, we allow that passes are assigned either to one antenna or none:

$$
\sum_{k\in C_i} y_{ik}\le 1,\quad \forall i\in F. \tag{21}
$$

Note the difference between this constraint and (1), where all passes must be assigned to exactly one antenna. Similarly subpasses can be rejected as well

$$
\sum_{\ell,k} y_{i\ell k}\le 1,\quad \forall i. \tag{22}
$$

Note the difference between this constraint and (14), where for each pass exactly one subpass must be selected. The remaining constraints and cost functions are as in Section 3.1 and Section 3.2.

In the example presented in Fig. 1, we could force a cancellation if we assume, as before, that pass 1 is compatible only with antenna 1 and pass 3 is compatible only with antenna 2, but now we disallow shortening. Thus constraints (5)–(11) are reduced to

$$
\begin{aligned}
&y_{11}\le 1,\\
&y_{21}+y_{22}\le 1,\\
&y_{32}\le 1,\\
&y_{11}+y_{21}\le 1,\\
&y_{22}+y_{32}\le 1.
\end{aligned}
$$

The cost index (12) remains the same. The obvious solution is to cancel either pass 1 or pass 2; since we have assumed they have the same priority, both solutions are equally valid. If not, one would cancel the lowest priority pass.

4. Computational results

This section aims at showing how our procedures are able to handle realistic instances, based on real schedules provided by the company, in just tens of seconds. Feedback from the company on preliminary test results were used to obtain appropriate values for the parameters ξ_{ik} and γ in (4) and (13), respectively.

The values of ξ_{ik} determine the relative weights between remaining in the original antenna ($e_{ik}=1$), changing antenna in the same site ($e_{ik}=0$, A_k in the same site), and changing site ($e_{ik}=0$, A_k in a different site). Initially, the company wanted to remain as close as possible to the original schedule, and insisted on having the parameters ξ_{ik} as described in (4), thus penalizing changes of the original antenna. However, from preliminary results we found that if the value of ξ_{ik} when changing antenna in the same site was relaxed (we set $\xi_{ik}=0.99$ instead of $1/2$ in that case), there were indeed less cancellations, while at the same time the solution was still keeping the passes in their original antennas most of the times. Not much was gained from other values. The value of 0.25 (when $e_{ik}=0$, A_k in a different site) was not changed for two reasons

1. this parameter does not have an important effect, as there are not many opportunities in practice to change site (given that different basses are on different locations and satellites usually do not fly over both of them on the same orbit),
2. the company tries to avoid changing site as much as possible, as this considerably modifies the timing of the pass.

The value of the parameter γ depends on whether one is more interested in keeping passes with their original length, or in maximizing the time that the antennas are actively used. We found that for the company a balance between the two objectives was appropriate, and choosing an intermediate value of $\gamma = 0.2$ gave satisfactory results for them. Choosing a larger γ results in more cancellations at the expense of the longer passes. Choosing a smaller γ results in selection of shorter passes when shortening, which is not desirable, without really improving the number of cancellations.

Next we present the results on 10 real instances containing about 3000 passes (a typical quantity of requests for a busy week), see Table 1. The usual number of satellites and antennas is 50 and 20, respectively and, on average, only 22% of the passes are marked as *shortable*. We note that in some cases the data correspond to challenging examples with a rather large number of satellites and sites involved in the conflicts. The manual resolution of these challenging examples by a skilled operator takes about one entire day of work. In Table 1, “Passes” refers to the number of passes considered, “Antennas” is the number of antennas, “satellites” is the number of satellites, “shortable passes” is the percentage of passes that are marked by the customer as shortable, “Conflicts” represents the number of conflicts arising in the corresponding instance, “Cancellations” is the total number of canceled passes, “Shortenings” represents the number of passes that were shortened, “Movements” refers to the number of movements done in each instance (showing between parenthesis movements to other sites), “Variables” and “Constraints” represent the number of variables and constraints in the corresponding ILP problem, respectively, and “Time” gives the time our solver required to solve the corresponding instance, in seconds.

These instances were built from real schedules for one week of requests. To perform the deconflicting operations and obtain an optimal deconflicted schedule, the following computational setup was used. The calculation of alternative passes from the antenna locations (and their admissible range of Azimuth-Elevation), and the satellite orbital elements (given as TLEs), was performed by the SaVoir software suite (http://www.taitussoftware.com), a powerful Satellite Planning and Mission Analysis tool. We developed (in C++) a software tool that efficiently computes all time intervals and possible subpasses following the steps of Section 2.4, and subsequently computes the problem constraints and coefficients of the cost function according to Section 3. To solve the resulting ILP problem, the free-software solver LPSolve was used. We have used version 5.5.2.0 for windows 32 bits (http://lpsolve.sourceforge.net/5.5/). All experiments were run on a laptop, Intel Core Duo P8400 2.26 GHz 3.00 Gb RAM, O.S. Windows 7 Professional 32 bits.

From the results cast in Table 1 we can affirm that the algorithm proposed in this project provides, for practical effects, an almost-*real-time* (around a minute) optimal solution for problems of considerable size (up to 4000 passes) that typically correspond to a full week of operation, according to the rules of operation set by the company. We note that, on average, 2605.2 passes were analyzed in each instance with 287.9 conflicts. The corresponding problems had 9154.7 variables and 9013.7 constraints on average. The average computational time required to solve these problems was 64.8 seconds. Only 3.13% of passes were canceled, 0.02% of passes were shortened, and 19.32% of passes were moved to other antennas (with 0.38% being moved to another site). Note however that many of the passes did not allow for shortening. We also observe that the computation time does not correlate directly with the size or the number of conflicts but depends more on the complexity of the conflicts.

Conversations with the company representatives let us know that the performance of our procedures exceeded the operator's

Table 1
Deconflicting results for 10 problem instances obtained from real-world examples. (Ant. = number of antennas, Sat. = number of satellites, Cancell. = number of cancellations, Short. = number of shortenings, Move. = number of movements, Vars = number of variables and Constr. = number of constraints.)

Passes	Ant.	Sat.	Shortable passes (%)	Conflicts	Cancell.	Short.	Move. total (other site)	Vars.	Constr.	Time (s)
3356	22	49	22	537	116	1	839 (13)	8090	9322	64
3006	22	47	22	196	75	3	703 (5)	11 219	12 157	61
3356	22	49	22	231	94	0	517 (6)	12 465	12 245	73
3566	22	50	22	306	114	1	561 (11)	12 360	12 714	73
3470	22	52	22	253	100	0	557 (17)	12 016	12 788	80
3408	22	52	22	196	91	0	478 (5)	12 289	12 025	71
1573	14	47	24	68	0	0	487 (7)	7143	4467	64
384	20	46	22	33	21	0	101 (0)	1247	1075	22
1586	16	45	24	6	2	0	479 (2)	7892	4964	63
2287	18	55	26	1053	250	4	312 (35)	6820	8380	77

expectations in terms of speed and quality of solutions with respect to their previous manual system. To be more concrete, they typically required 1 or 2 days to find a solution, whereas our algorithm takes about one minute. This feature was extremely interesting for them, as a fast algorithm can be used to research the effect on performance of prospective locations of future antennas or to test the capacity of the network to accept more customers. Additionally, we were able to diminish the number of cancellations given the global perspective of the algorithm; we obtained about 2 to 4 fewer cancellations in each scenario.

5. Conclusions

In this paper, we have introduced Integer Linear Programming models to efficiently manage the scheduling of passes for a multi-antenna, multi-site ground network serving numerous satellite operators. The aim of our methods is to solve conflicts in the best possible way while respecting preferred assignments and priorities. By following the rules and principles of operation of the ground network company, and developing simple models of deconflicting, we have been able to feasibly solve challenging scenarios, finding exact solutions in short times with an open-source solver. Operational constraints and priorities have been efficiently integrated in the modeling and different adjustments can be achieved by tuning the cost weights according to the specifications of the passes.

Our models have been tested over a number of realistic instances provided by the ground network operator, which was previously scheduling the passes manually. Conversations with the company representatives let us know that the performance of our procedures exceeded the operator's expectations in terms of speed and quality of solutions (few number of movements, even fewer number of cancellations) with respect to their previous manual system.

Among future possible refinements, we could mention the inclusion of additional objectives, such as fairness criteria (penalizing multiple cancellations for the same customer) or the development of advanced tools such as adaptive online scheduling, which would imply a scheduler running online with capabilities such as including last-minute requests for passes as they come, or immediately adapting to dynamically changing constraints, such as antenna failures. Another possible line of future research is to try to exploit the similarities between our problem and the disjunctive scheduling problem in order to find suitable models and algorithms.

Conflict of interest statement

For the numerical experiments we have made use of real data provided by Kongsberg Satellite Services AS (KSAT) with authorization.

By the non-disclosure agreement conditions we have not identified the name and location of the antennas, satellites and customers.

Acknowledgements

The authors gratefully acknowledge the cooperation of Taitus Software (http://www.taitussoftware.com) and its team, and in particular its founder and CEO (Felipe Martín Crespo), which introduced this problem to us and provided integration with its orbital mechanics visual software SaVoir. We also acknowledge the cooperation of Kongsberg Satellite Services AS (KSAT). Jorge Galán acknowledges financial support through grants MTM2012-31821 and P12-FQM-1658. Federico Perea acknowledges financial support through grants FQM-5849 and P09-TEP-5022 (Junta de Andalucía and FEDER) and MTM2010-19576-C02-01 and MTM2013-46962-C2-1 (MICINN, Spain). Special thanks are due to three anonymous referees for their valuable comments.

References

[1] M.A. Aloulou, C. Artigues, Flexible solutions in disjunctive scheduling: general formulation and study of the flow-shop case, Comput. Oper. Res. 37 (2010) 890–898.
[2] L. Barbulescu, A.E. Howe, J.P. Watson, L.D. Whitley, Satellite range scheduling: a comparison of genetic, heuristic and local search, in: Lecture Notes in Computer Science, vol. 2439, 2002, pp. 611–620.
[3] L. Barbulescu, J.-P. Watson, L. Whitley, A. Howe, Scheduling space–ground communications for the air force satellite control network, J. Sched. 7 (2004) 7–34.
[4] L. Barbulescu, A. Howe, D. Whitley, AFSCN scheduling: how the problem and solution have evolved, Math. Comput. Model. 43 (2006) 1023–1037.
[5] L. Barbulescu, A.E. Howe, L.D. Whitley, M. Roberts, Understanding algorithm performance on an oversubscribed scheduling application, J. Artif. Intell. Res. 27 (2006) 577–615.
[6] E. Barret, D. Curtis, Introduction to Environmental Remote Sensing, 3rd edition, Springer, 1992.
[7] M. Bester, Automated multi-mission scheduling and control center operations at UC Berkeley, in: IEEE Aerospace Conference, 2009, pp. 1–12.
[8] S.A. Chien, M. Johnston, J. Frank, M. Giuliano, A. Kavelaars, C. Lenzen, N. Policella, A generalized timeline representation, services, and interface for automating space mission operations, in: The 12th International Conference on Space Operations, Stockholm, Sweden, June 2012.
[9] B. Clement, M.D. Johnston, in: A. Press (Ed.), The Deep Space Network Scheduling Problem, IAAI, Pittsburgh, PA, USA, 2005.
[10] G. Corrao, R. Falone, E. Gambi, S. Spinsante, Ground station activity planning through a multi-algorithm optimisation approach, in: IEEE First AESS European Conference on Satellite Telecommunications (ESTEL), Rome, Italy, 2012.
[11] H.D. Curtis, Orbital Mechanics for Engineering Students, 2nd edition, Butterworth-Heinemann, 2009.
[12] U. Dorndorf, E. Pesch, T. Phan-Huy, Constraint propagation techniques for the disjunctive scheduling problem, Artif. Intell. 122 (2000) 189–240.
[13] J. Galán-Vioque, R. Vázquez, E. Carrizosa, C. Vera, F. Perea, F. Martín, Towards a visual tool for swath acquisition planning in multiple-mission EOSs, in: IWPSS 2011 Workshop Proceedings, Darmstad, Germany, 2011.
[14] S. Lee, W.C. Jung, J.-H. Kim, Task scheduling algorithm for the communication, ocean, and meteorological satellite, ETRI J. 30 (1) (2008) 1–12.
[15] F. Marinelli, S. Nocella, F. Rossi, S. Smriglio, A lagrangian heuristic for satellite range scheduling with resource constraints, Comput. Oper. Res. 38 (2011) 1572–1583.
[16] K. Schilling, Earth observation by distributed networks of small satellites, in: International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2009.
[17] M. Schmidt, K. Schilling, A scheduling system with redundant scheduling capabilities, in: Third IEEE International Conference on Space Mission Challenges for Information Technology, Pasadena, California, USA, July 2009.
[18] M. Schmidt, M. Rybysc, K. Schilling, A scheduling system for small ground station networks, in: SpaceOps 2008 Conference Hosted by ESA and EUMETSAT in Association with AIAA, 2008.
[19] P. Torres, P. Lopez, On not-first/not-last conditions in disjunctive scheduling, Eur. J. Oper. Res. 127 (2000) 332–343.
[20] D. Vallado, W. McClain, Fundamentals of Astrodynamics and Applications, 3rd Edition, Microcosm Press/Springer, 2007.
[21] F. Xhafa, X. Herrero, A. Barolli, L. Barolli, Evaluation of struggle strategy in genetic algorithms for ground stations scheduling problem, J. Comput. Syst. Sci. 79 (2013) 1086–1100.
[22] Z. Zhang, N. Zhang, Z. Feng, Multi-satellite control resource scheduling based on ant colony optimization, Expert Syst. Appl. 41 (2014) 2816–2823.
[23] N. Zufferey, P. Amstutz, P. Giaccari, Graph colouring approaches for a satellite range scheduling problem, J. Sched. 11 (2008) 263–277.