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

ARTICLE INFO

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

ABSTRACT

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.