Managing SOHO's Keyhole Periods: Problem Definition and Solving Model

Nicola Policella and **Henrique Oliveira**
European Space Operations Center
European Space Agency
Darmstadt, Germany
name.surname@esa.int

Tero Siili
SOHO Team
Goddard Space Flight Center
European Space Agency
Greenbelt, MD, USA
name.surname@esa.int

Abstract

This work introduces a combinatorial problem which arises in the joint ESA/NASA program SOHO. This mission concerns the generation of high-quality instrument data recording and downlinking schedules for the so-called Keyhole Periods. While the main goal is to maximize the science data return, the problem is characterized by several kinds of constraints, such as on-board memory capacity, limited communication windows over the downlink channel, need for robustness against losses of communication opportunities as well as personnel resources.

The contribution of this paper is twofold: on one hand it provides a description of the problem together with a timeline-based representation, and on the other it introduces a solving approach based on a flow network model of the problem and a Max-Flow-based solving procedure.

Introduction

The Solar and Heliospheric Observatory (SOHO) is an ESA/NASA mission to observe the Sun and the Solar wind. It was launched on 2 Dec 1995 and inserted to its elliptical so-called halo orbit around the L1 Sun-Earth Lagrangian point (approximately 1.5 million km sunwards from Earth) in Feb 1996. This special orbit was chosen to permit continuous, uninterrupted view of the Sun — not achievable with Earth-orbiting observatories. The spacecraft weighed 1850 kg at launch, the largest dimension of the bus is approximately 4.3 m and the solar arrays span 9.5 m (Fig. 1). SOHO is a three-axis stabilized spacecraft keeping its Sun-observing instruments constantly facing the Sun with high pointing accuracy. The payload comprises 12 instruments, most of which are operating, despite of the mission having exceeded its 2-year design lifetime by more than a decade. The mission is operated primarily from NASA Goddard Space Flight Center (GSFC) near Washington DC by a team comprising ESA, NASA, contractor and science instrument personnel. Some instruments are operated remotely from the teams' home institutes. Communication links with SOHO are provided primarily by NASA's Deep Space Network (DSN). For more details of the SOHO mission, see (Domingo, Fleck, & Poland 1995; Fleck *et al*. 2006).

Copyright © 2009, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

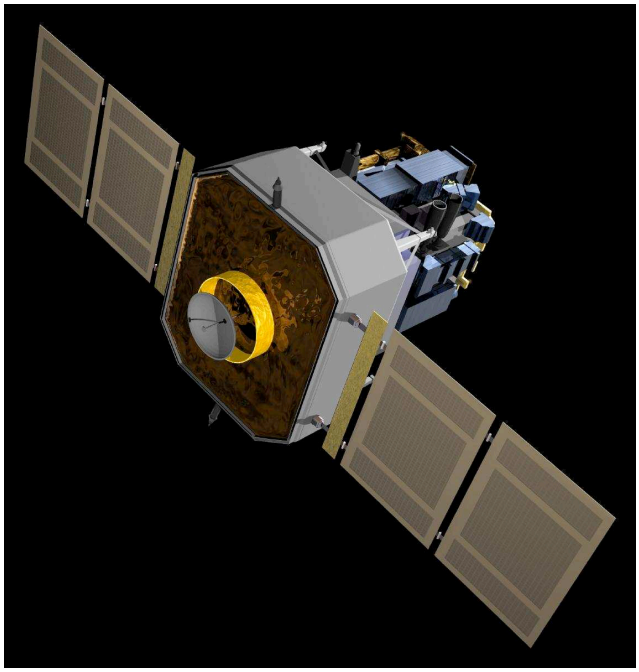

Figure 1: SOHO spacecraft. Image courtesy of SOHO (ESA & NASA).

The problem considered here concerns generation of plans and schedules for SOHO data recording and downlinking during periods of reduced telemetry capabilities. These special periods arose later in the mission out of a problem with SOHO's High-Gain Antenna (HGA) and occur four times per year, lasting approximately 2-4 weeks each. The periods are called Keyholes, hence the problem is referred to as the Keyhole Periods Problem. The ultimate goal is to generate plans which maximize the amount of science data returned to the ground under a set of several constraints.

Similar problems can arise in satellite domains such as the ones described in (Bensana, Lemaitre, & Verfaillie 1999; Verfaillie & Lemaitre 2001). Both works concern a set of Earth observation operations to be allocated over time under a set of mandatory constraints such as: no overlapping images, sufficient transition times (or *setup* times), bounded

instantaneous data flow and on-board limited recording capacity. In (Knight 2006) instead the author focuses his attention on the case of swath segment (i.e., fixed trajectory observations) scheduling for orbiting spacecraft, investigating different algorithms to obtain optimal solutions.

An example, even more related to our problem, is the MEXAR2 project (Cesta *et al*. 2007a; 2007b): this tool is used to synthesize the operational commands for data downlink from the on-board memory of an interplanetary space mission spacecraft to the ground stations.

Our approach starts from the Max-flow based algorithm previously proposed by (Oddi & Policella 2007) and then we extend it to cope with the specific characteristics of the Keyhole Periods Problem. Both problems refer to the management of the on-board memory of a spacecraft. Notwithstanding, in our case we have the additional goal of deciding when and where the recording activities have to be executed – these production activities were instead already allocated and given as input in the problem considered in (Oddi & Policella 2007).

The paper starts by giving on overview of the SOHO mission. The background of the Keyhole Periods problem and the solving approach are described. Finally, the current status and future plans of the project are outlined.

What is a keyhole?

The term *keyhole* stems from radar antenna terminology in the context of coverage areas and maps thereof. A radar can not "see" in a circular area in the immediate vicinity of the antenna. Obstructions (such as hilltops and buildings) near the antenna will also cause wedge-shaped areas of no radar coverage. These patterns drawn on aerial maps resemble the shape of a keyhole. Use of the term has since been generalised to refer to any area not covered by an antenna. In the context of SOHO, keyholes are periods of intermittent or reduced telemetry occurring four times per year and lasting 2-4 weeks each. The keyholes are caused by the limited pointing range of SOHOs High-Gain Antenna (HGA).

Figure 2 shows SOHO's orbit, indicating the HGA geometry. To get a good downlink signal, SOHO's High Gain Antenna (HGA) needs to be pointed correctly as the spacecraft traverses its highly elliptical orbit around the L1 Lagrangian point, some 1.5 million km sunward of the Earth. Originally SOHO's HGA with its two degrees of freedom was able to be continually pointed towards the Earth and the Deep Space Network's (DSN) receiving antennas throughout the halo orbit and to provide a high-data-rate telemetry capability.

In 2003, however, HGA mechanism problems arose which eventually lead to the decision to abandon moving HGA along one of the two degrees of freedom (Fleck *et al*. 2006). This limited the pointing capability. Together with the requirement to keep the spacecraft oriented towards the Sun resulted in turn to two portions of the halo orbit, in which the HGA with its only one degree of freedom can no longer be turned to an attitude, where it would "see" the Earth and provide high-rate telemetry (Fig. 2). These portions of the orbit are the Keyholes and they occur twice per orbit. The period of the orbit around the L1 is about 6 months, hence keyholes occur roughly every 3 months. The

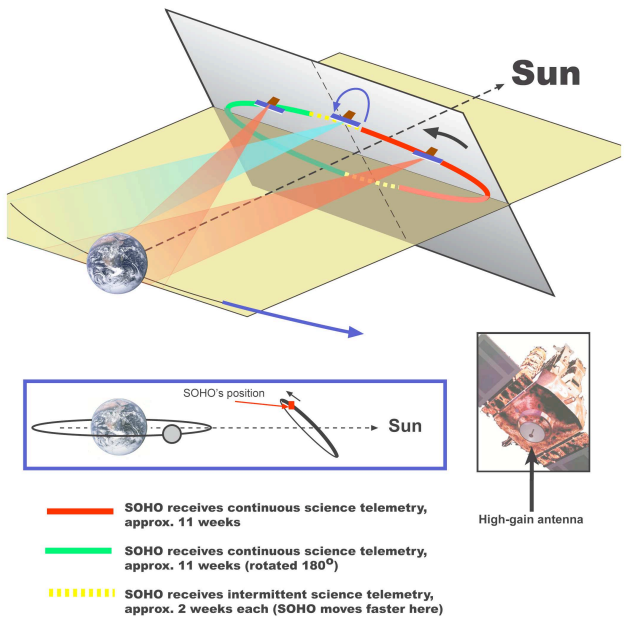

Figure 2: Schematic of SOHO's orbit, indicating the geometry of the High Gain Antenna. Image courtesy of SOHO (ESA & NASA).

HGA location in the fixed direction of movement was chosen to minimize the keyhole durations.

26-m keyhole	34-m keyhole	26-m keyhole
High rate with HGA and 34 m and 70 m antennas	High rate with LGA and 70 m antennas	High rate with HGA and 34 m and 70 m antennas

Table 1: Keyhole period (2-4 weeks)

In non-keyhole parts of SOHOs halo orbit DSNs 26-m and 34-m antennas can provide continuous high-rate telemetry via the HGA. During keyhole periods high-rate downlinks can be provided by antennas as described in Table 1. This table shows that, at the start and end of a keyhole period, both 34m and 70m antennas may be used for data dumping. In the middle, however, only 70m antennas are available. During gaps in contact the data are stored in SOHO's on-board recorders and high-rate telemetry is required to *dump* or empty the recorders. An example Keyhole schedule is shown in Fig. 3. Note that each contact (communication interval) is followed by a no-communication interval and vice versa.

As an example, Fig. 3 shows the November 2008 keyhole period. The "white" sections represent gaps in contact between SOHO and the DSN antennas, whereas the passes (contacts to DSN antennas) are identified by red or blue sections. During all passes instrument data are downlinked in real time. *Dumping* (emptying of recorders) is possible only through some DSN antennas: these are identified by the

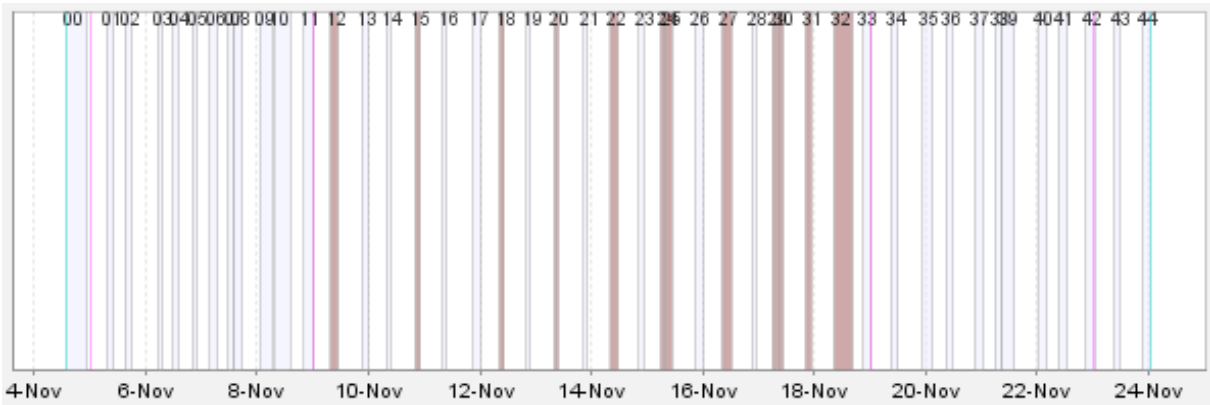

Figure 3: November 2008 SOHO Keyhole Period. White sections represent gaps in contact between SOHO and the DSN antennas. Red sections are contacts with DSN antennas enabling real-time data downlink only, blue sections are contacts with antennas with both real-time and dumping downlinks.

"blue" sections. The "red" sections instead only real-time data can be downloaded (therefore during these periods the amount of data in the recorders will not change). Finally, it is worth remarking that in the overall keyhole period each communication interval is followed by a no-communication interval and vice-versa.

Data production and on-board data storage

During real-time contacts data of SOHO's all twelve instruments are downlinked. During gaps the data are stored on two storage devices, a Solid-State Recorder (SSR) and a Tape Recorder (TR) for dumping during subsequent contacts. These have different capacities (the SSR has double the size of TR), and while the SSR is a random-access device, the TR is sequential in nature. Both SSR and TR are LIFO (i.e. dumping is done by downlinking the data in reverse). The TR's filling rate is fixed and it always records the full data complement, whereas SSR's filling rate can be adjusted to a subset of the full complement.

During Keyhole periods, the availability of DSN 70-m antennas (Table 1) is not sufficient to downlink all data of all instruments, hence an Intermittent Recording Patch was developed for the SSR. In essence this modification allows the SSR to record the data of selected and prioritized subsets of the full 12-instrument complement. Since the data rates of SOHO's instruments vary, each subset has a different total data rate. This permits both adjustment of the data flow as well as prioritization of instruments, whose data are chosen for recording. In the current mission phase six instrument *subsets* have been implemented.[1] These are identified by lists of SOHO instrument one-letter identifiers.[2]

[1]In addition to the subsets, also so-called *submodes* cause the SSR filling rate to vary. This aspect is not further considered here; readers may refer to the following link for further details: http://sohowww.nascom.nasa.gov/soc/submode.html.

[2]For instance, VGM stands for VIRGO, GOLF and MDI, whereas VGMFL would be VIRGO, GOLF, MDI, CELIAS (= F) and LASCO.

Keyhole planning process: goals and current approach

Given the schedule of the available antenna passes, the planning process consists of selecting the optimal instrument subset(s) for each gap such that the data return is maximized, while the constraints are adhered to. The constraints include mandatory requirements for a timeseries as uninterrupted as possible for a selected subset of instruments,[3] limited on-board data storage, limited availability and durations of communication windows, maximum data rate in communication links and a typically constant data production rate. Other softer, nice-to-have, requirements include the need for robustness of the plan against pass losses and the need to avoid complicated and labor-intensive ground operations tasks when executing the plan. The adjustable quantity is the recorded data (sub)set — which instruments' data are recorded and when.

In the current manual approach, the keyhole plan is a text file, which initially contains descriptions of the passes (generated from DSN schedule files). Currently start and end times of the keyhole subperiods (Table 1), instrument subset selections, swaps between the SSR and TR, timings of recorder dumps and commands as well as instructions to the Flight Operations Team are added to the file with a text editor by the human planner. The modified plan is then run through the Interactive Data Language (IDL) based keyhole-planning tool,[4] which analyzes the plan, provides warnings if constraints are not adhered to, as well as metrics/statistics and a graphical display/representation of the plan for the planner's evaluation.

[3]In practice, it is not possible to set the recorder to "idle", always a subset has to be selected. In the current SOHO mission phase, the highest-priority criterion is to have continuous retrieval of at least the helioseismology subset VGM – this is considered the minimum acceptable data-rate subset and constitutes the "VGM constraint" to planning.

[4]This tool has been developed internally by the SOHO team and is currently used to support the testing and evaluation phase of the approach proposed in this paper.

The "human" planner *walks* through the keyhole time period in an iterative fashion and modifies it, until a safe and hopefully high-data-return plan is achieved. This process takes for a new keyhole plan 1-3 workdays (depending on the length and complexity of the DSN schedule). Of course changes in DSN schedules, pass losses and unexpected observational opportunities lead to further adjustments of the plan.

The current approach works, but it has a learning curve for the "human" planner, who in effect carries out the optimization process based on experience. The current keyhole planning tool also provides neither an initial (perhaps coarsely optimized) selection of subsets nor what-if type parallel solutions for the planner. One need is to have a system in place capable of generating (semi-)automatically such proposals for evaluation and either approval or further changes and fine-tuning by the human planner.

Model-based Representation with Timelines

The modeling core is based on the temporal evolution of key components and on the ability to capture relevant domain constraints. This approach to the solution based on "timeline synthesis" for problem components, is common to solid works in space domain such as, RAX-PS/EUROPA (Jonsson *et al*. 2000), ASPEN (Chien *et al*. 2000), and MEXAR2 (Cesta *et al*. 2007a).

The timeline based approach focuses on the main and relevant problem features. In particular it considers the temporal evolution of specific system components. In particular in the case of the SOHO Keyhole periods problem two types of relevant components have been identified: the recorders, and the transmission channel.

This modeling approach allows for reducing the problem to temporal functions representing the amount of data manipulated over time by these key components, such that the constraints given by the instruments filling rate, the recorder capacity, and the channel bandwidth are satisfied. In the remainder of this section we discuss the details of the timeline-based model whereas in the next sections, we introduce a Max-Flow reduction of such a representation that enabled the use of standard Max-Flow algorithms as solvers.

Model variables and constraints. Figure 4 shows the model we designed for the SOHO Keyhole Periods problem. This consists of three timelines representing three different components: SSR recorder, TR recorder, and the communication channel. Based on these three timelines the model is refined as follows:

1. The temporal horizon has been subdivided in contiguous intervals $\{t_1, t_2, \ldots, t_n\}$, such that the different communication/no-communication periods are considered. Each of these intervals, t_j, is characterized by the following two values:
   - dur_{t_j}, the duration of the interval t_j;
   - $rate_{t_j}$, the downlink data rate of the interval (0 if t_j is a no-communication interval).

   We note that both the previous values are fixed for each interval t_j.

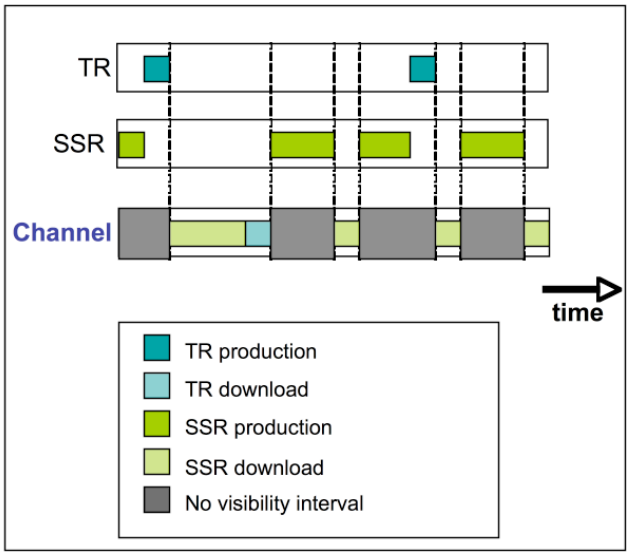

Figure 4: Timeline model

2. Two different *decision variables* are then associated to each interval and each of the two recorders (SSR and TR):
   - d_{ij}, each variable represents the amount of data recorded during the time interval t_j to the recorder i,
   - δ_{ij}, each variable represents the volume of data dumped within each time slot t_j from the recorder i.

   We observe that the second variables represent activities on both the associated recorder and the channel timeline while the d_{ij} variables represent activities on the related recorder timelines. For the sake of the discussion, in the following sections we will assume that recorders 1 and 2 are respectively SSR and TR.

A further aspect still to be discussed is the instruments filling rate. As discussed before, this can in general vary from interval to interval due to the different filling rates of the SSR subsets. Therefore its natural representation would be a further decision variable. Conversely we decided to model this aspect by considering, for each interval, the maximum filling rate, and introducing a constant $fill_{max}$ to represent this value. In the next sections we will show how the proposed solving approach is able to retain the flexibility given by the presence of SSR subsets.

Having introduced the variables we end the discussion introducing the constraints present in the problem. We have the following local constraints:

$$d_{ij} \begin{cases} \leq fill_{max} \times dur_{t_j} & \text{if } rate_{t_j} = 0 \\ = 0 & \text{elsewhere} \end{cases} \quad (1)$$

$$\delta_{ij} \begin{cases} \leq rate_{t_j} \times dur_{t_j} & \text{if } rate_{t_j} > 0 \\ = 0 & \text{elsewhere} \end{cases} \quad (2)$$

The first constraint refers to the maximum amount of data that can be produced during a time interval, while the second

constraint refers to the maximum amount of data that can be downloaded. We also remark that it is not possible to have on the same interval data recording (d_{ij}) and data dumping ($\delta_{ij} > 0$). In fact, during visibility period the real-time flow of data coming from the several instruments is directly transmitted to the ground station without need of recording it.

Over the above variables we also have three types of global constraints:

- Production Capacity – as the instruments have a fixed production rate, given a time interval, it is possible to calculate the maximum amount of data that can be produced (see above). Moreover, as the two recorders are sharing the same instruments the sum of the data stored in the two recorders cannot be greater that the "Production Capacity", i.e.:

$$\sum_i d_{ij} \leq fill_{max} \times dur_{t_j} \quad \forall t_j \tag{3}$$

- Recorder Capacity – as mentioned above, both the SSR and the TR have limited storage capacities, cap_i. In order to avoid data loss, the cumulative flow of data for each recorder (both data stored and downloaded) must not be grater than its capacity:

$$0 \leq \sum_j (d_{ij} - \delta_{ij}) \leq cap_i \quad \forall i, j \tag{4}$$

- Channel capacity – the whole amount of data downlinked over a given time period must not exceed the channel's capacity, i.e.:

$$\sum_i \delta_{ij} \leq rate_{t_j} \times dur_{t_j} \quad \forall t_j \tag{5}$$

Due to the flexibility of the flow network model, these constraints are relatively simple to express. The robustness requirement mentioned above is also included in the model in this fashion. Recorder capacity must be preserved to save data produced by the highest priority subset, in case one of the passes is lost.

Solving Approach

In this section we introduce a formalization for the Keyhole Periods Problem as a Max-Flow problem and the flow network associated to. As mentioned before this approach extends the method introduced in (Oddi & Policella 2007).

Flow Networks and the Max-Flow Problem

Before the introduction of the flow network model for the Keyhole Periods Problem, in the following we briefly review the theory behind the Max-Flow problem (Cormen *et al.* 2001). A flow network $G(V, E)$ is a direct graph where V is a set of vertices and E is a set of edges (u, v) with nonnegative capacity $c(u, v) \geq 0$. The flow network has two special vertices: a source s and sink t. A flow in G is a integer-valued function $f : V \times V \rightarrow \mathbb{Z}$ (we consider only integer-valued flows) that satisfies the following three properties:

- capacity constraint: for all $u, v \in V$,
$$f(u, v) \leq c(u, v)$$
- skew symmetry: for all $u, v \in V$,
$$f(u, v) = -f(v, u)$$
- flow conservation: for all $u \in V \cup \{s, t\}$,
$$\sum_{v \in V} f(u, v) = 0.$$

The quantity $f(u, v)$ can be positive or negative, and it represents the *net flow* from vertex u to vertex v. The value of a flow f into the graph G, is defined as

$$f = \sum_{v \in V} f(s, v),$$

that is the total flow out of the source. In the Max-Flow problem given a flow network G, the goal is to find a flow of maximum value from source to sink.

The use of Max-Flow supports different flexible aspects particularly useful for our problem. For example, it finds, whenever exists, a consistent solution given any initial situation, e.g., the initial situation can be empty or not (some dumps and/or recording activities already allocated). This property is important for example to enable the user to impose some initial operations. Hence it is possible to use the algorithm to complete the timelines when a solution exists.

Flow-network model

Figure 5 shows an example of flow network. There are five types of nodes: source, sink, filling nodes f_j, recorder nodes r_{ij}, and channel nodes c_j.

Each of the internal nodes (i.e., filling, recorder, and channel) represents one of the constraints discussed in the previous section (Fig. 6 shows the partial flow network for a particular time interval t_j). In particular we have:

- The recorder node is used to model (4); this can be obtained considering the flow conservation property, and setting to cap_i the capacity of the arcs between two recorder nodes (i.e., $f(r_{ij}, r_{i(j+1)}) \leq cap_i$).
- The filling node permits to consider the amount of data produced in the interval t_j; the flow through the arc between the source and the filling node is equal to the sum of the data produced over the two recorders, $\delta_{1j} + \delta_{2j}$. Therefore, setting to $fill_{max} \times dur_{t_j}$ the capacity of the arc between the source and the filling node (i.e., $f(source, f_j) \leq fill_{max} \times dur_{t_j}$) we have (3) represented.
- Symmetrically, the channel node is used for representing (5): it is in fact sufficient to set to $rate_{t_j} \times dur_{t_j}$ the capacity of the arc between the channel node and the sink (i.e., $f(c_j, sink) \leq rate_{t_j} \times dur_{t_j}$).

In the current discussion we assumed an initial empty value for each recorder; to represent situations with a different initial value, it would be sufficient (as already applied in (Oddi & Policella 2007)) to extend the flow network with a set of arcs from the source to each of the recorder nodes, labeled with a capacity value equals to the initial value of the data stored in each recorder.

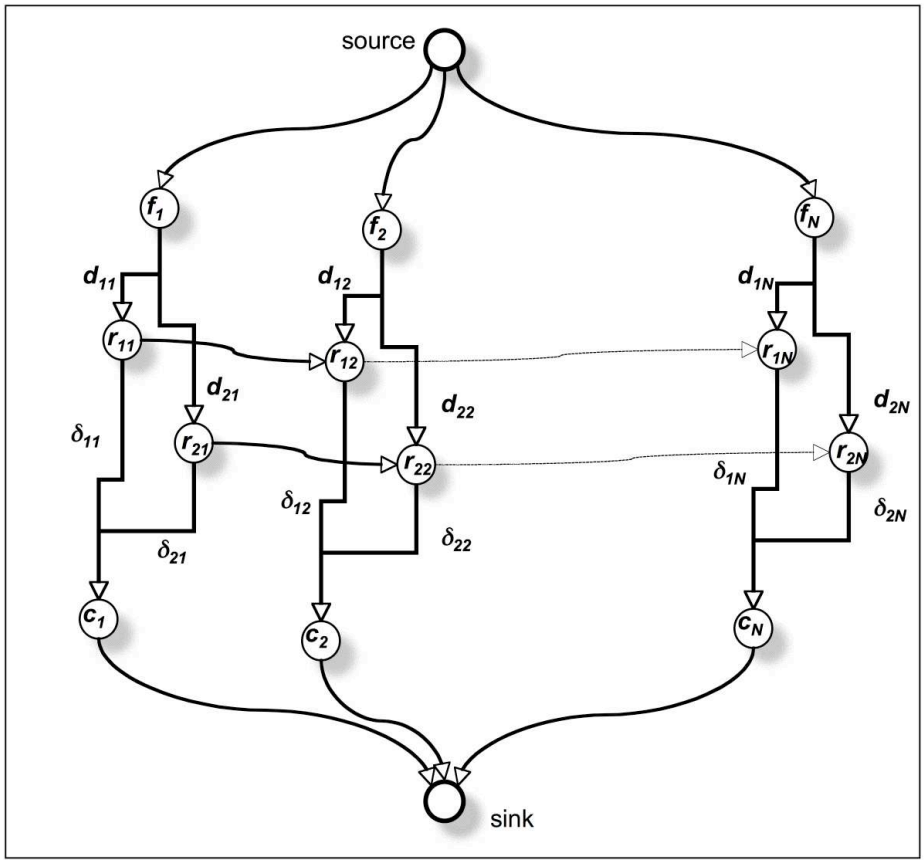

Figure 5: Flow network model

Solving Method

As aforementioned, to find a solution to a Keyhole Periods Problem, it is sufficient to apply a max-flow algorithm to the associated flow network (see Fig. 5), and then the set of recorder operations and downlink activities can be obtained with a simple procedure. Before discussing the details of the plan generation, it is worth remarking that usually a max-flow problem has more than one solution. Our current approach consists in generating the solution which minimizes the waiting time for the data return. This is obtained by modifying the Edmonds-Karp algorithm (Edmonds & Karp 1972) for generating maximum flow (as previously exploited in (Oddi & Policella 2007)). Basically, this algorithm identifies the maximum flow in a depth-first way. Readers interested on a throughout discussion about Flow-networks and Max-flow algorithms can find an essential survey in (Cormen *et al*. 2001).

In the rest of the section we discuss how, given a max-flow solution, we can synthesize a plan for a Keyhole Period. Given a Max-Flow solution, a solution to the Keyhole problem is obtained by setting the previous decision variables as follows:

$$d_{ij} = f(f_j, r_{ij})$$

and

$$\delta_{ij} = f(r_{ij}, c_j).$$

From these assignments we can obtain straightforwardly the following activities (see timeline model in Fig. 4):

Downlink activities: given a time interval t_j and the two related variables δ_{ij}, we have two downlink activities, one for each recorder, of duration $\delta_{ij}/rate_{t_j}$.

TR recording activities: given a time interval t_j and the related TR variables d_{2j}, we have a TR recording activity of duration $d_{2j}/fill_{max}$.

What is missing is the SSR recording activities. In the next section we discuss how to obtain these activities starting from the value d_{1j}.

Reintroducing SSR subsets. In a previous section we described how the SSR allows the selection of different subsets of the on-board instruments. This allows a more flexible recorder with different filling rates available (six in our case). In this section we will exploit the SSR flexibility for synthesizing the recording activities. In particular, we define $\{rate_1, rate_2, \ldots, rate_k\}$ as the ordered set of the rate values of the different SSR subsets (i.e., $rate_l < rate_{(l+1)}$ and $rate_k = fill_{max}$).

From the discussion above we know that one of the inputs of this phase is the value d_{1j}, i.e., the amount of data stored in SSR during the time interval t_j. Another input value is the time frame dedicated to recording on SSR, $t_{j,ssr} = t_j -$

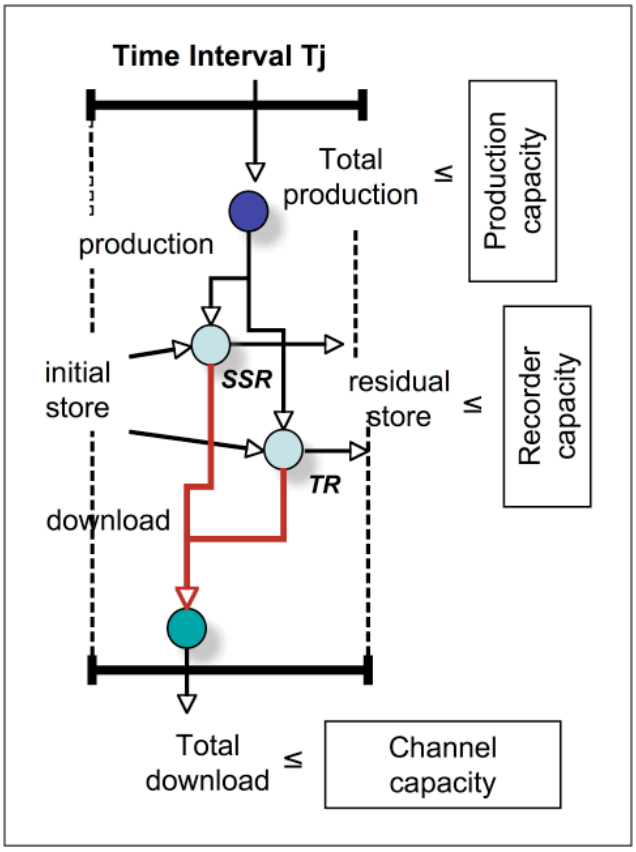

Figure 6: Flow network model for one time interval

$d_{2j}/fill_{max}$. Finally, an implicit constraint of this phase requires to maximize the coverage of the time interval $t_{j,ssr}$ with SSR recording activities.

Given the above input values we define the *ideal* filling rate as $rate^* = d_{1j}/t_{j,ssr}$. Comparing this value with the set of filling rates, we may have three different cases:

- $rate^* < rate_1$ – This is the worst case, in fact the amount of data planned on SSR is not sufficient to cover the whole time frame, not even with the lowest rate subset. Therefore we have a **single SSR recording activity** with the lowest rate subset and duration $d_{1j}/rate_1 < t_{j,ssr}$.
- $\exists l \ s.t. \ rate^* = rate_l$ – This is the simplest case, in fact, we have a **single SSR recording activity** with the subset associated to $rate_l$ and duration $t_{j,ssr}$.
- $\exists l \ s.t. \ rate_l < rate^* < rate_{(l+1)}$ – As Fig. 7 shows, in this case we synthesize two activities. Even though it is possible to synthesize a single activity with rate $rate_l$ and duration $t_{j,ssr}$, this would lead to a reduced value of data return. Therefore we have **two SSR recording activities** with, respectively, rate[5] $rate_l$ and $rate_{(l+1)}$, and

[5]Of course the order of the subsets can be changed.

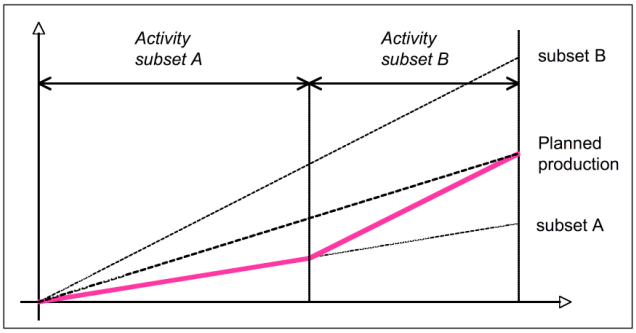

Figure 7: Generating two SSR recording activities

durations[6]

$$t_{j,ssr} - \frac{d_{1j} - rate_l \times t_{j,ssr}}{rate_{(l+1)}} \text{ and } \frac{d_{1j} - rate_l \times t_{j,ssr}}{rate_{(l+1)}}.$$

The case $rate^* > rate_k$ is not possible due to the presence of the Production capacity constraint (3). It is also worth observing that in all the three previous cases, the amount of data synthesized by the max-flow approach, d_{1j}, is preserved, maintaining the solution's quality.

SOHO versus Mars Express case

In (Oddi & Policella 2007) the authors refer to a problem in the domain of the Mars Express (MEX) mission. In the latter the memory is organized with several independent modules or packet stores. Also a set of Payload Operation Requests (POR), which have the effect of storing data in the memory, is given as input. The main goal is to produce a schedule for downloading the stored data.

Conversely, in the context of SOHO, there are no PORs since all relevant instruments are on all the time (and produce data). Also SOHO has (in MEX terminology) only two packet stores (SSR and TR). No orbit events like MEX factor in to the planning process as the only events are the actual passes available (real-time data vs. playback). For SOHO and for the keyhole planning, robustness is resilience against the disappearance/loss of a pass – a plan safety criterion is satisfaction of the VGM constraint even in case of loss of individual dump-capable passes.

To summarize, both problems share the need of managing the on-board memory of a spacecraft. However, in the case of SOHO we have the additional goal of deciding *when* and *where* the recording activities have to be executed – in the case of the MEX problem these production activities were instead already allocated and given in input. The need of also synthesizing the recording activities results in increased complexity in the case of the Keyhole Periods Problem. To face this additional aspect we have extended the solving approach, both redesigning the flow network model and adding an ad-hoc procedure for extracting, given a Max-Flow solution, a final plan.

[6]The values can be obtained through basic geometry operations.

Current status and future work

A prototype tool, named SKEYP, based on the Max-Flow approach described here is currently under development to become part of the *SOHO Keyhole planning toolbox* together with the current IDL-based tool. For the sake of the discussion, in this paper we have omitted several "low level" details. These have been considered in order to design an operational tool.

An even more important aspect is the need to cope with schedule and other changes and uncertainties. As mentioned before, the initial plan may become invalid after changes of the initial inputs of the problem (e.g., DSN availability). The goal will be to produce solutions which are either robust to these changes or easily adjustable. In particular the idea we are investigating is to use the flexibility of the flow-network model to produce solutions which might better satisfy operational requirements. This would potentially facilitate and speed up testing and implementation of keyhole plan changes — currently this needs to be done manually by the planner using the IDL tool.

As part of the current activities, we planned a thorough evaluation of this new approach. Different aspects may be considered such as the mere performances of the tool (e.g., quality of the solutions), time to produce a solution, and the usability of the tool (user viewpoint). The latter aspect is also important if we consider the need of training new possible operators. An intelligent and easy to use tool will simplify the handover phases between operators.

Conclusions

In this paper we introduced and discussed the Keyhole Periods Problem; in the context of the SOHO mission, keyholes are periods of intermittent or reduced telemetry occurring four times per year and lasting 2-4 weeks each. The keyholes are caused by the limited pointing range of SOHOs High-Gain Antenna (HGA). A solution to this problem consists in synthesizing a set of recording activities (to store data during no-communication intervals) and a set of downlink operations (to dump the stored data).

The paper describes a Max-Flow based approach to solve this problem. This approach has been inspired by the method described in (Oddi & Policella 2007), in fact both problems refer to the management of the on-board memory of a spacecraft. The main difference concerns the need of deciding when and where the recording activities have to be executed, whereas, these activities were already defined and given in input in the problem considered in (Oddi & Policella 2007). We have discussed how the problem can be modeled by using a flow network, and how a solution can be extracted from a Max-Flow solution for the modeling flow network.

References

Bensana, E.; Lemaitre, M.; and Verfaillie, G. 1999. Earth Observation Satellite Management. *Constraints: An International Journal* 4(3):293–299.

Cesta, A.; Cortellessa, G.; Denis, M.; Donati, A.; Fratini, S.; Oddi, A.; Policella, N.; Rabenau, E.; and Schulster, J. 2007a. MEXAR2: AI Solves Mission Planner Problems. *IEEE Intelligent Systems* 22(4):12–19.

Cesta, A.; Cortellessa, G.; Fratini, S.; Oddi, A.; and Policella, N. 2007b. An innovative product for space mission planning – an *a posteriori* evaluation. In *ICAPS-07. Proceedings of the 17th International Conference on Automated Planning & Scheduling*, 57–64.

Chien, S.; Rabideau, G.; Knight, R.; Sherwood, R.; Engelhardt, B.; Mutz, D.; Estlin, T.; Smith, B.; Fisher, F.; Barrett, T.; Stebbins, G.; and Tran, D. 2000. ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling. In *Proceedings of the 6th International Conference on Space Operations, SpaceOps 2000*.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2001. *Introduction to Algorithms,* Second Edition. MIT Press.

Domingo, V.; Fleck, B.; and Poland, A. 1995. *The SOHO mission: an overview*. Kluwer. 1–37. Reprinted from *Solar Physics, Volume 162, Nos. 1–2, 1995*.

Edmonds, J., and Karp, R. M. 1972. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the Association for Computing Machinery* 19(2):248–264.

Fleck, B.; Müller, D.; Haugan, S.; Duarte, L. S.; and Siili, T. 2006. 10 years of SOHO. *ESA Bulletin* 126:24–32.

Jonsson, A.; Morris, P.; Muscettola, N.; Rajan, K.; and Smith, B. 2000. Planning in Interplanetary Space: Theory and Practice. In *Proceedings of the Fifth Int. Conf. on Artificial Intelligence Planning and Scheduling (AIPS-00)*.

Knight, R. 2006. Solving Swath Problems Optimally. In *2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06)*.

Oddi, A., and Policella, N. 2007. Improving Robustness of Spacecraft Downlink Schedules. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 37(5):887–896.

Verfaillie, G., and Lemaitre, M. 2001. Selecting and Scheduling Observations for Agile Satellites: Some Lessons from the Constraint Reasoning Community Point of View. In Walsh, T., ed., *Principles and Practice of Constraint Programming, 7th International Conference, CP 2001*, number 2239 in Lecture Notes in Computer Science, 670–684. Springer.