AIMS: A Tool for Long-term Planning of the ESA INTEGRAL Mission*

Cédric Pralet and **Gérard Verfaillie**
ONERA, 2 av. Édouard Belin, BP 74025, F-31055 Toulouse Cédex 4, France
{Cedric.Pralet,Gerard.Verfaillie}@onera.fr

Abstract

This paper presents a tool called *AIMS* for *APSI Integral Mission Scheduler* dedicated to the automatic yearly selection and scheduling of the observations to be performed by the *ESA INTEGRAL* satellite, a satellite in charge of the observation of gamma-ray emissions from the universe. More precisely, the paper describes the main features of the problem to manage, a constraint-based model of this problem, the specific local search algorithm which has been developed to solve it on top of the planning and scheduling core *APSI* framework, and the resulting *AIMS* tool. It shows the gains that may result from the use of such a tool with regard to a previous manual management of the problem.

1. Introduction

Two years ago, the European Space Agency (*ESA*) initiated a study called *APSI* for *Advanced Planning and Scheduling Initiative* whose objective was to explore the use of Artificial Intelligence Planning and Scheduling technologies for the management of *ESA* missions. More precisely, the study aimed at developing a generic framework allowing planning and scheduling problems to be modeled and solved and at experimenting this framework on three planning problems coming from current or future *ESA* missions. The consortium in charge of the study should include one industrial partner and three European research centers.

In the selected consortium, the industrial partner was *VEGA* (Darmstadt, Germany) and the research centers were *ISTC-CNR* (Rome, Italy), *Politecnico di Milano* (Milan, Italy), and *ONERA* (Toulouse, France). *ISTC-CNR* was in charge of the development of the generic planning and scheduling framework we will refer to as the core *APSI* framework in the paper. *ISTC-CNR* was also in charge of the first case study: a pre-planning of the maintenance and communication activities in the context of the *ESA Mars Express* mission. *ONERA* was in charge of the second case study: the long-term planning of the observation activities of the *ESA INTEGRAL* satellite. Finally, *Politecnico di Milano* was in charge of the third case study: the long-term planning of the observation activities of the *ESA XMM-Newton* satellite. See (Steel et al. 2009) for more details about the *APSI* study.

This paper is dedicated to the second case study. In Section 2, we describe the problem that the *INTEGRAL* Science Operations Centre (*ISOC*) from the European Space Astronomy Center (*ESAC*) sets upon us. In section 3, we show how we modeled this problem as a constrained optimization problem, using variables, domains, constraints, and criteria. In Section 4, we describe the specific local search algorithm we developed to solve it on top of the core *APSI* framework. Section 5 describes the resulting *AIMS* tool. We conclude with several lessons which could be drawn from this work.

2. The INTEGRAL long-term planning problem

The *INTEGRAL* (*INTErnational Gamma-Ray Astrophysics Laboratory*) mission aims at observing gamma-ray emissions from the universe using several instruments on board an Earth-orbiting satellite. It is an *ESA* mission, managed in cooperation with Russia and the USA. Starting in 2002 for at least two years, it has been extended until 2012 (see http://www.esa.int/esaMI/Integral/).

The *INTEGRAL* satellite embeds four observation instruments: *SPI* (a gamma-ray spectrometer), *IBIS* (a gamma-ray imager), *Jem-X* (an X-ray monitor), and *OMC* (an optical monitor camera). These four instruments are fixed on the platform and pointing in the same direction (see Figure 1). The *AOCS* system allows the platform and thus the instruments to remain pointed in a given direction during an observation and to move from a direction to another one between two successive observations. Reaction wheels are used for attitude control.

The satellite is moving on a highly elliptical orbit around the Earth. One revolution takes 72 hours and, among them, only about 58 hours, out of the Earth radiation belts, are available for observation. Due to the presence of the Sun, the Earth, the Moon, and other planets, a given target is not permanently observable during these 58 hours. For each target and each satellite revolution, observation windows can be pre-computed.

The observation of a given target requires a given number of observation patterns to be performed. There are four classes of patterns: rectangular patterns requiring 25 point-

*The work presented in this paper has been performed in the context of the *ESA APSI* study (*Advanced Planning and Scheduling Initiative*) whose partners are *VEGA* (Darmstadt, Germany), *ISTC-CNR* (Rome, Italy), *Politecnico di Milano* (Milan, Italy), and *ONERA* (Toulouse, France).

Figure 1: **The INTEGRAL satellite**

ings, hexagonal patterns requiring 7 pointings, staring patterns requiring only one pointing, and user-defined patterns. Each pointing takes a given duration. It is not mandatory and it is often impossible to perform an observation within one revolution. As a consequence, an observation can be split into sub-observations, each one including a chosen number of patterns. It is even not mandatory to perform an observation pattern in one shot: rectangular, hexagonal, and user-defined patterns can be split into sub-patterns, each one including a chosen number of pointings. We assume that, for each observation, the user defines the number N of elementary observations which make it up and thus their common duration: the observation duration divided by N.

With each observation of a given target, is associated a type which specifies the way observation must be performed. There are four observation types:

- normal observations (*NO*) which must be split as less as possible and ended as early as possible after they started;
- no-splitting observations (*NS*) which must not be interleaved with other observations;
- periodic observations (*PE(p,t)*) which must be decomposed into elementary observations to be performed every p revolutions with a tolerance of t on the deviation from the period;
- spread observations (*SP(n)*) which must be decomposed into n sub-observations to be spread as much as possible over the year and to be each performed with no splitting.

The slewing time between two successive observations is not directly considered in the long-term planning. It is however indirectly considered by limiting the number of different observations within each revolution. Moreover, in order to keep some time available for opportunistic observations of unexpected events such as the appearance of new X-ray/gamma-ray sources, only a given percentage of the observation time within each revolution is considered to be available at the planning time. Constraints related to energy, data recording and downloading are not taken into account in the long-term planning.

Each year, the *ISOC* emits an announcement of opportunity (*AO*) to which scientists answer by emitting observation requests. Then, a target allocation committee selects observation requests and assigns to each selected request a priority and a realization percentage above which the observation is considered to be achieved (in general, 100% is not mandatory).

The long-term planning problem consists in selecting and scheduling over the next observation period (generally of one year duration) the observations associated with the current *AO*. Note that in practice all the observations associated with an *AO* cannot be fully satisfied over one observation period: the problem is over-constrained. As a consequence, the observations that are associated with the previous *AO*, but can be only partially satisfied over the current observation period, must be added (with a high priority) to those associated with the current *AO*.

Throughout the observation period, regular re-planning is necessary due to unexpected observation requests or to changes in the existing observation requests. Moreover, the long-term plan serves as an input for regular short-term planning which decides on the detailed activities to be performed by the satellite.

The resulting long-term planning problem is a kind of over-constrained scheduling problem (Smith 2004; Kramer, Barbulescu, and Smith 2007) where the objective is to perform each observation as much and as well as possible, taking observation priorities into account and knowing that each observation has at its disposal a set of observation windows and can be split into sub-observations, but that observations cannot overlap.

3. A constraint-based model

3.1 Data

The problem data is the following one:

- a set R of revolutions over the planning horizon and, for each revolution $r \in R$, a starting time $S(r)$ and an ending time $E(r)$ of the window available for observation within r, as well as a maximum filling percentage $M(r)$ of r; windows associated with two revolutions do not overlap;
- a set P of priority levels and for each level $p \in P$, a weight $WE(p)$ which reflects the relative importance of observations of level p.
- a set O of observations and for each observation $o \in O$, a type $TY(o)$ (NO for *normal*, NS for *no-splitting*, $PE(p,t)$ for *periodic*, or $SP(n)$ for *spread*), a priority level $P(o) \in P$, a total duration $D(o)$, the duration $DEO(o)$ of an elementary observation, a number $NEO(o)$ of elementary observations ($D(o) = NEO(o) \cdot DEO(o)$), a percentage $PCA(o)$ above which o is considered to be achieved, and a set $W(o)$ of windows available for o; for each observation window $w \in W(o)$, a

revolution $R(w) \in R$, a starting time $S(w)$ and an ending time $E(w)$; several windows may be available for performing an observation o within a revolution r; however, they do not overlap; if $TY(o) = SP(n)$, observation o is decomposed into a set $SO(o)$ of n sub-observations and the set $W(o)$ of the windows associated with o is partitioned into n subsets, each one $W(so)$ associated with a sub-observation $so \in SO(o)$.

In order to model the problem, we introduce the notion of observation activity: an observation activity is a set (possibly empty) of contiguous elementary observations associated with an observation o in a window w.

With each normal observation o ($TY(o) = NO$) and each window $w \in W(o)$, we systematically associate two observation activities. This allows more interleaving between observations and is very useful when some of the observations must be performed at a fixed time. However, with each special observation o ($TY(o) \in \{NS, PE, SP\}$) and each window $w \in W(o)$, we associate only one observation activity: for no-splitting observations, no interleaving with other observations is allowed; for periodic observations, only one elementary observation is allowed per revolution; finally, for spread observations, each sub-observation is no-splitting.

In the sequel, OA denotes the set of observation activities. For each $oa \in OA$, $O(oa)$ denotes the associated observation, $W(oa)$ the associated window, and $R(oa)$ the associated revolution. For each observation $o \in O$ (resp. each revolution $r \in R$), $OA(o)$ (resp. $OA(r)$) denotes the set of observation activities associated with o (resp. r).

To the problem data, must be finally added the maximum number MOA of no empty observation activities per revolution, useful to limit time spent on slewing between observation activities.

Figure 2 shows an example of solution (plan) of an instance involving 5 revolutions and 8 observations. For example, for observation o_1, we have 6 observation windows and 4 non empty observation activities: the first one involving 3 elementary observations in the first window and the other three ones involving each 2 elementary observations in the last three windows. For observation o_2, we have 4 observation windows and 4 non empty observation activities, each one involving only one elementary observation in each window. We can observe two non empty observation activities in the same window for observation o_3 in the last window. We can also observe that no observation activity is associated with observations o_5 and o_6 (all of them are empty). Finally, we can observe that the duration of an elementary observation is observation dependent: for example, greater with o_2 than with o_1.

3.2 Variables

Given this data, the problem is to choose for each observation activity $oa \in OA$:

- its starting time $s(oa) \in [S(W(oa)), E(W(oa))]$;
- the number $neo(oa) \in [0..NEO(O(oa))]$ of elementary observations it involves.

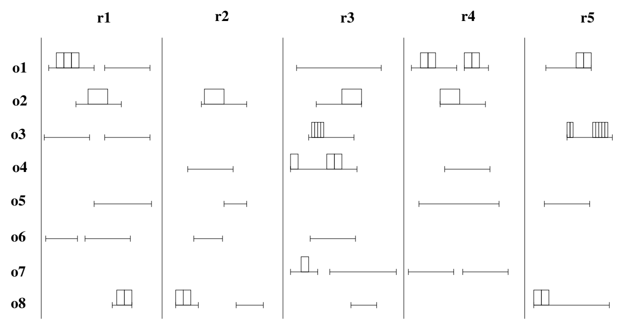


Figure 2: **Example of solution (plan)**

Other variables, which are functional expressions of the previous ones, may be useful to express constraints.

- for each observation activity $oa \in OA$, its ending time $e(oa) = s(oa) + neo(oa) \cdot DEO(O(oa))$ with the following domain: $e(oa) \in [S(W(oa)), E(W(oa))]$;
- for each observation $o \in O$, the number $neo(o) = \sum_{oa \in OA(o)} neo(oa)$ of performed elementary observations, with the following domain: $neo(o) \in [0..NEO(o)]$.

For each observation o, let $seq(o)$ be the sequence of non empty observation activities associated with o, ordered according to their starting time. Let $foa(o)$ (resp. $loa(o)$) be the first (resp. last) observation activity in $seq(o)$ (equal to $\emptyset$ if $seq(o) = \emptyset$). For each observation activity oa in $seq(o)$, let $noa(oa)$ be the next one in $seq(o)$ (equal to $\emptyset$ if $oa = loa(o)$).

3.3 Constraints

In addition to the domains of variables $s(oa)$, $neo(oa)$, $e(oa)$, and $neo(o)$, the following constraints must be satisfied.

For each revolution r, no overlapping between observation activities associated with r:

$$\begin{aligned} \forall r \in R, \quad & \forall oa, \forall oa' \in OA(r) \,|\, oa \neq oa' : \\ & ((neo(oa) > 0) \wedge (neo(oa') > 0)) \rightarrow \\ & ((e(oa) \leq s(oa')) \vee (e(oa') \leq s(oa))) \end{aligned} \tag{1}$$

Note that we can express these constraints separately for each revolution, because windows associated with two revolutions do not overlap.

For each revolution r, maximum filling percentage of r:

$$\begin{aligned} \forall r \in R : \quad & \sum_{oa \in OA(r)} neo(oa) \cdot DEO(O(oa)) \\ & \leq M(r) \cdot (E(r) - S(r)) \end{aligned} \tag{2}$$

For each revolution r, maximum number of non empty observation activities in r[1]:

[1] As usual in Constraint Programming tools, a constraint c is considered equivalent to a boolean variable b equal to 1 if c is satisfied and 0 otherwise (constraint reification).

$$\forall r \in R : \sum_{oa \in OA(r)} (neo(oa) > 0) \leq MOA \quad (3)$$

For each no-splitting observation o, no hole in the sequence of non empty observation activities associated with o:

$$\begin{aligned} &\forall o \in O \quad | \, (TY(o) = NS), \quad \forall oa \in OA(o) : \quad (4) \\ &\quad (neo(o) > 0) \rightarrow \\ &\qquad ((s(foa(o)) < s(oa) < s(loa(o))) \\ &\qquad \rightarrow (neo(oa) > 0)) \end{aligned}$$

For each no-splitting observation o, no interleaving with any other observation:

$$\begin{aligned} &\forall o \in O \quad | \, (TY(o) = NS), \; \forall oa \in (OA - OA(o)) : \; (5) \\ &\quad (neo(o) > 0) \rightarrow \\ &\qquad ((s(foa(o)) < s(oa) < s(loa(o))) \\ &\qquad \rightarrow (neo(oa) = 0)) \end{aligned}$$

For each periodic observation o of periodicity p and of tolerance t, only elementary periodic observation activities:

$$\begin{aligned} &\forall o \in O \quad | \, (TY(o) = PE(p,t)), \quad \forall oa \in OA(o) : \; (6) \\ &\quad ((0 \leq neo(oa) \leq 1) \wedge \\ &\quad (((neo(oa) > 0) \wedge (noa(oa) \neq \emptyset)) \rightarrow \\ &\qquad ((R(noa(oa)) > R(oa)) \wedge \\ &\qquad (|R(noa(oa)) - R(oa) - p| \leq t)))) \end{aligned}$$

For each spread observation o, maximum number of elementary observations associated with each sub-observation:

$$\begin{aligned} &\forall o \in O \mid (TY(o) = SP(n)), \quad \forall so \in SO(o) : \; (7) \\ &\sum_{oa \in OA(o) \mid W(oa) \in W(so)} neo(oa) \leq NEO(o)/n \end{aligned}$$

Note that, for any observation o, the maximum number of associated elementary observations is enforced by the domain of $neo(o)$.

3.4 Criteria

As usual in most of the real world applications, the definition of the optimization criterion is not as easy as the definition of the constraints is, mainly because of the presence of several not well defined conflicting criteria. After interaction with the *ISOC*, we adopted the following definition of the optimization criterion.

For each observation o, let $q(o)$ be the quality associated with o in the current plan. The global criterion q to be maximized is defined as the normalized weighted sum of observation qualities:

$$q = \frac{\sum_{o \in O} WE(o) \cdot q(o)}{\sum_{o \in O} WE(o)} \quad (8)$$

For each observation o, let $qc(o)$ be the completion quality of o and $qr(o)$ be its realization quality. Let $\alpha \in [0, 1]$ be a user defined constant which expresses the trade-off between observation completion and realization. The quality $q(o)$ associated with o is defined as the weighted sum of completion and realization qualities of o:

$$\forall o \in O : q(o) = \alpha \cdot qc(o) + (1 - \alpha) \cdot qr(o) \quad (9)$$

For each observation o, the completion quality $qc(o)$ depends on the percentage of completion of o:

$$\begin{aligned} &\forall o \in O : \quad (10) \\ &qc(o) = \begin{cases} \frac{neo(o)}{NEO(o) \cdot PCA(o)} \text{ if } (\frac{neo(o)}{NEO(o)} \leq PCA(o)) \\ 1 \text{ otherwise} \end{cases} \end{aligned}$$

The realization quality $qr(o)$ of an observation depends on its type.

For each normal or no-splitting observation o, the objective is to finish o as early as possible after it started. Thus, the realization quality depends on the distance (in terms of revolutions) between the last and the first non empty observation activity associated with o. Let $\delta(o) = R(loa(o)) - R(foa(o))$ be this distance. Let $\Delta_{max}(o)$ be the maximum value of $\delta(o)$, obtained when the first and last observation activities associated with o are both non empty. Let $\delta_{min}(o)$ be the minimum value of $\delta(o)$ which could be obtained if the plan would contain only $neo(o)$ elementary observations associated with o and no observation activity associated with any other observation. $\Delta_{max}(o)$ can be precomputed and $\delta_{min}(o)$ can be pre-computed for each possible value of $neo(o)$.

$$\begin{aligned} &\forall o \in O | \; (TY(o) \in \{NO, NS\}) : \quad (11) \\ &qr(o) = \begin{cases} 0 \text{ if } (neo(o) = 0) \\ 1 \text{ if } ((neo(o) \neq 0) \wedge (\delta_{min}(o) = \Delta_{max}(o))) \\ \frac{\Delta_{max}(o) - \delta(o)}{\Delta_{max}(o) - \delta_{min}(o)} \text{ otherwise} \end{cases} \end{aligned}$$

For each periodic observation o, the objective is to satisfy as well as possible the periodicity constraint. Thus, the realization quality depends on the sum of the deviations from the period. Let $sd(o) = \sum_{oa \in OA(o) | ((neo(oa) > 0) \wedge (noa(oa) \neq \emptyset))} |R(noa(oa)) - R(oa) - p|$ be this sum.

$$\begin{aligned} &\forall o \in O | \; (TY(o) = PE(p, t) : \quad (12) \\ &qr(o) = \begin{cases} 0 \text{ if } (neo(o) = 0) \\ 1 \text{ if } (neo(o) = 1) \\ 1 - \frac{sd(o)}{(neo(o) - 1) \cdot t} \text{ otherwise} \end{cases} \end{aligned}$$

Finally, for each spread observation o, its realization quality is defined as the mean value of the realization qualities of its associated no-splitting sub-observations:

$$\begin{aligned} &\forall o \in O \quad | \, (TY(o) = SP(n) : \quad (13) \\ &qr(o) = \frac{\sum_{so \in SO(o)} qr(so)}{n} \end{aligned}$$

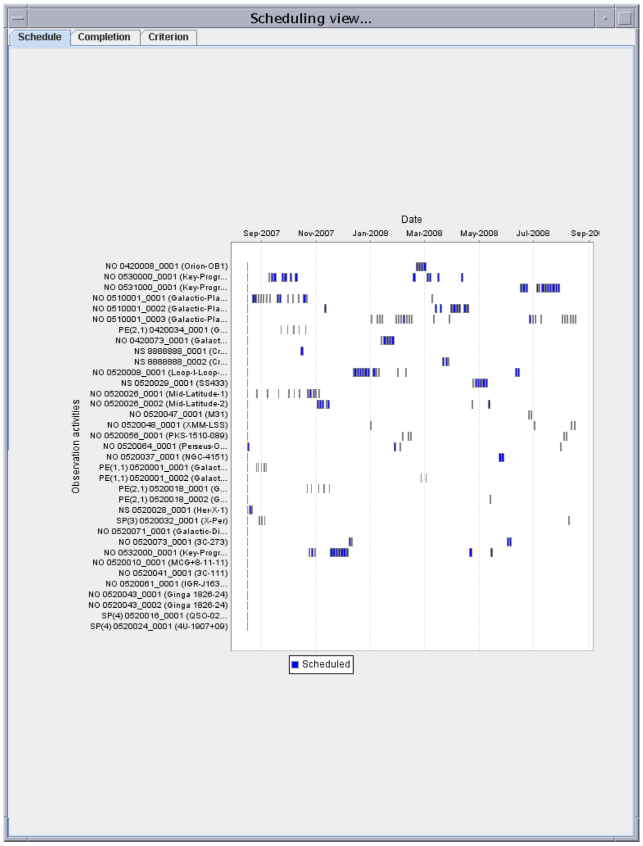


Figure 3: **Current plan, with all the non empty observation activities**

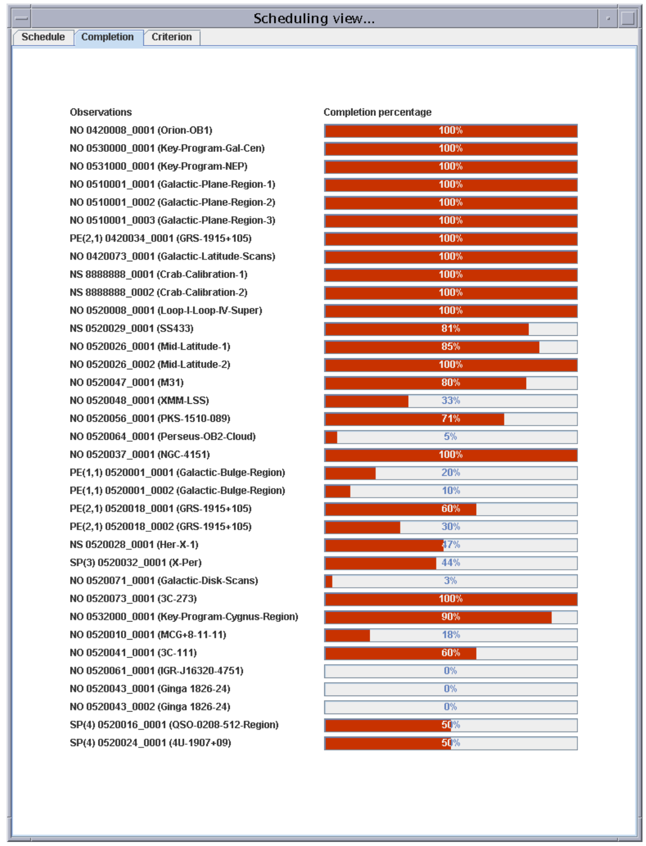


Figure 4: **Current completion percentage of each observation**

All the defined qualities q, $q(o)$, $qc(o)$, and $qr(o)$ take their value between 0 and 1.

3.5 Problem size

The instance associated with the *AO* we worked on (covering period from August 2007 to August 2008) involves 123 revolutions and 35 observations, which generate 2731 observation activities: one or two per pair made of an observation and a possible associated window. The number of elementary observations associated with an observation is between 1 and 1023, with most of the observations requiring several hundreds of elementary observations.

As a consequence, if we consider only variables $neo(oa)$, we get 2731 variables whose domain size is between 2 and 1024.

4. A local search algorithm

The main features of the algorithm we designed and implemented are the following ones:

- it is a local search algorithm; such a choice is justified by the size of the instances to be solved which precludes the use of complete optimal algorithms, such as a systematic tree search;
- the algorithm systematically starts from an empty plan, which is obviously consistent; it maintains a current plan and it modifies it iteratively using two kinds of local move: either the enlargement of an observation activity oa (by adding to oa as many as possible elementary observations), or the enlargement of an observation activity oa after the removal of another non empty observation activity oa' in the same revolution (by removing all the elementary observations associated with oa' and then adding to oa as many as possible elementary observations); at each step of the algorithm, consistency of the current plan is maintained: all the constraints presented in Section 3.3, from 1 to 7, are satisfied;
- at each step of the algorithm, only a small subset of the set of possible local moves is pre-selected; pre-selection is necessary because of the huge number of observation activities and thus of possible local moves; evaluating each of them would dramatically decrease the number of local moves per time unit and thus the global algorithm performance; pre-selection uses a combination of heuristic and random choices (for example, to choose randomly a non empty observation activity to be removed among those of lowest priority);
- at each step of the algorithm, all the pre-selected local

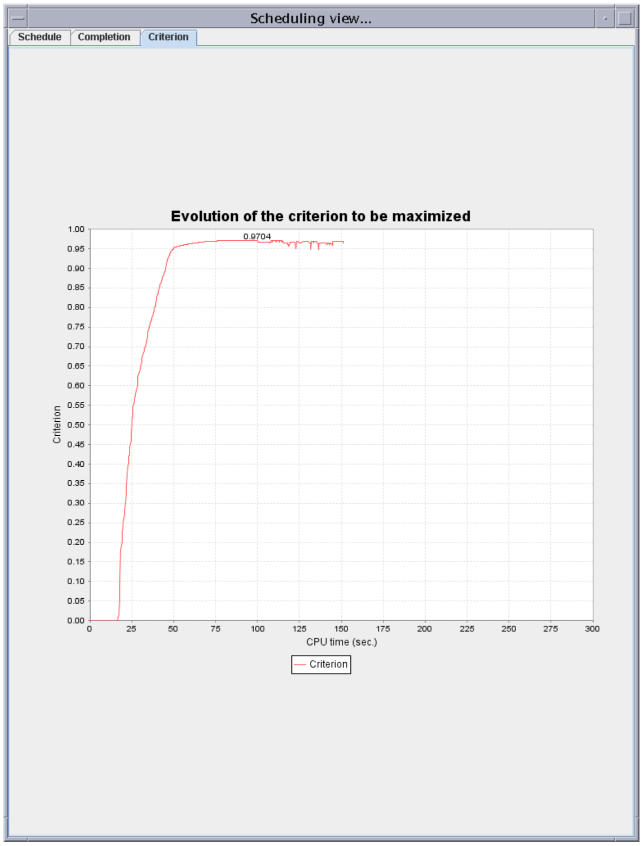
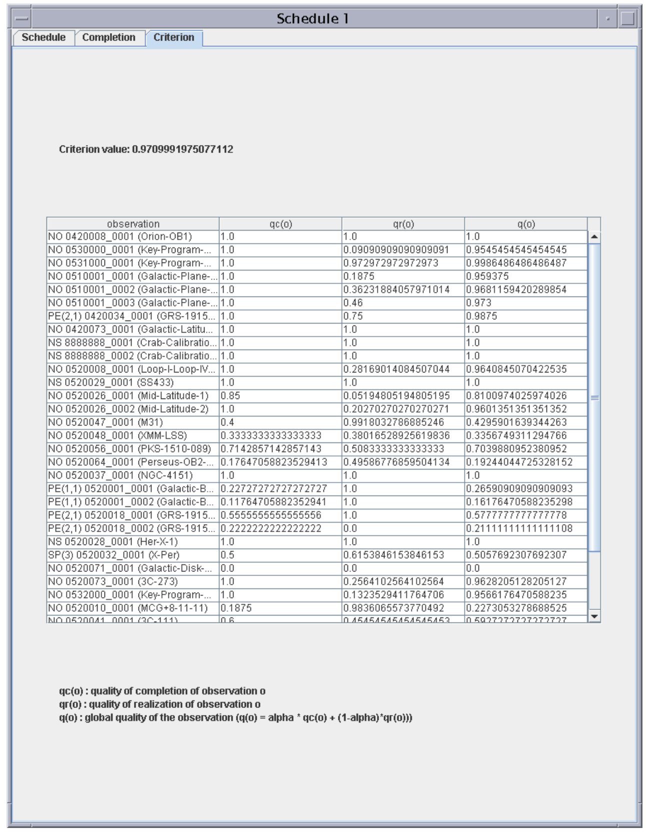


Figure 5: **Evolution of the plan quality and best quality found so far**



observation	qc(o)	qr(o)	q(o)
NO 0420008_0001 (Orion-OB1)	1.0	1.0	1.0
NO 0530000_0001 (Key-Program-...	1.0	0.09090909090909091	0.9545454545454545
NO 0531000_0001 (Key-Program-...	1.0	0.972972972972973	0.9986486486486487
NO 0510001_0001 (Galactic-Plane-...	1.0	0.1875	0.959375
NO 0510001_0002 (Galactic-Plane-...	1.0	0.36231884057971014	0.9681159420289854
NO 0510001_0003 (Galactic-Plane-...	1.0	0.46	0.973
PE(2,1) 0420034_0001 (GRS-1915...	1.0	0.75	0.9875
NO 0420073_0001 (Galactic-Latitu...	1.0	1.0	1.0
NS 8888888_0001 (Crab-Calibratio...	1.0	1.0	1.0
NS 8888888_0002 (Crab-Calibratio...	1.0	1.0	1.0
NO 0520008_0001 (Loop-I-Loop-IV...	1.0	0.28169014084507044	0.9640845070422535
NS 0520029_0001 (SS433)	1.0	1.0	1.0
NO 0520026_0001 (Mid-Latitude-1)	0.85	0.05194805194805195	0.8100974025974026
NO 0520026_0002 (Mid-Latitude-2)	1.0	0.20270270270270271	0.9601351351351352
NO 0520047_0001 (M31)	0.4	0.9918032786885246	0.4295901639344263
NO 0520048_0001 (XMM-LSS)	0.3333333333333333	0.38016528925619836	0.3356749311294766
NO 0520056_0001 (PKS-1510-089)	0.7142857142857143	0.5083333333333333	0.7039880952380952
NO 0520064_0001 (Perseus-OB2-...	0.17647058823529413	0.49586776859504134	0.19244044725328152
NO 0520037_0001 (NGC-4151)	1.0	1.0	1.0
PE(1,1) 0520001_0001 (Galactic-B...	0.22727272727272727	1.0	0.26590909090909093
PE(1,1) 0520001_0002 (Galactic-B...	0.11764705882352941	1.0	0.16176470588235298
PE(2,1) 0520018_0001 (GRS-1915...	0.5555555555555556	1.0	0.5777777777777778
PE(2,1) 0520018_0002 (GRS-1915...	0.2222222222222222	0.0	0.21111111111111108
NS 0520028_0001 (Her-X-1)	1.0	1.0	1.0
SP(3) 0520032_0001 (X-Per)	0.5	0.6153846153846153	0.5057692307692307
NO 0520071_0001 (Galactic-Disk-...	0.0	0.0	0.0
NO 0520073_0001 (3C-273)	1.0	0.2564102564102564	0.9628205128205127
NO 0532000_0001 (Key-Program-...	1.0	0.1323529411764706	0.9566176470588235
NO 0520010_0001 (MCG+8-11-11)	0.1875	0.9836065573770492	0.2273053278688525

Figure 6: **Completion, realization, and aggregate quality of each observation in a plan**

moves are evaluated via an estimation of their positive or negative impact on the plan quality; one of them is then randomly selected among those of highest estimated impact; the selected local move is effectively applied if the estimated impact is strictly positive and applied with a probability P if the estimated impact is negative or null;

- in order to avoid cycles around local optima, a tabu list of the T previous local moves is maintained; at each step of the algorithm, all the local moves that belong to the current tabu list cannot be selected;
- in order to diversify the search, the algorithm is regularly restarted from an empty plan; more precisely, it is restarted each time a maximum number L of local moves without any strict improvement in the quality of the best plan found is reached;
- the whole local search algorithm is built on top of the core *APSI* framework (Cesta and Fratini 2008) it uses as a subroutine in order to answer at each step of the algorithm questions such as "What is the maximum number of elementary observations that can be added to an observation activity oa within a revolution r, given all the other observation activities currently present within r?"; the core *APSI* framework is also used to maintain a flexible schedule of the observation activities in each revolution.

To sum up, the resulting algorithm performs a local search which starts from an empty plan and uses specific local moves, stochastic heuristics to select, to choose, and to accept local moves, a tabu list to avoid cycles in the sequence of local moves, and restarts to diversify the search. It tries to combine the best ideas present in state-of-the-art local search algorithms such as hill-climbing search, tabu search, and simulated annealing (Aarts and Lenstra 1997).

Inside the algorithm, the basic scheduling questions are managed by the core *APSI* framework which is able to reason in a generic way on tasks, temporal constraints, and resource constraints. However, other questions, such as the satisfaction of special constraints associated with special observations (no-splitting, periodic, and spread ones) or the optimization of the plan quality, are managed by the algorithm itself, outside from the core *APSI* framework.

5. The AIMS tool

The user can run the *AIMS* tool by specifying:

- the directory where the data are available;
- the number N of plans he/she wants; the tool will therefore provide him/her with the N best plans found;
- the maximum CPU time he/she is ready to spend;

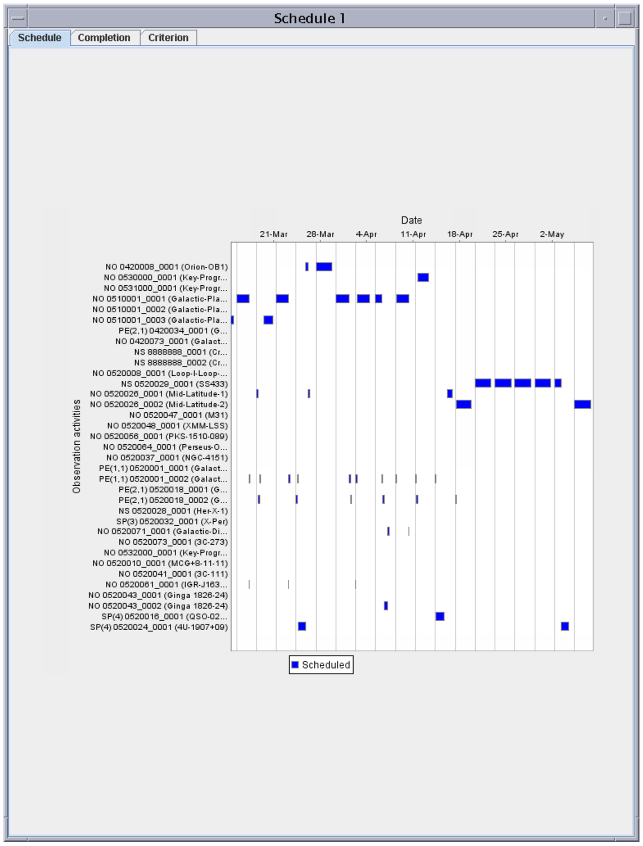


Figure 7: **Zoom on a part of a plan**

- the trade-off α between observation completion and realization quality (see Equation 9);
- the maximum number MOA of non empty observation activities per revolution (see Constraint 3).

An advanced user can play with other algorithmic parameters such as:

- the maximum number L of local moves without any strict improvement in the quality of the best plan found which triggers a restart of the algorithm from an empty plan;
- the length T of the tabu list;
- the probability P of acceptance of a local move which does not increase the plan quality.

After starting the algorithm, he/she can follow its evolution by looking at three windows:

- a first one which shows the current plan with all the non empty observation activities (see Figure 3);
- a second one which offers another view of the current plan by showing the completion percentage of each observation (see Figure 4);
- the third one which shows the evolution of the plan quality and the best quality found so far (see Figure 5).

In the first two figures, observations are ordered from top to bottom by decreasing priority.

In the third figure, we can observe an increasing quality phase which corresponds to an initial filling of the plan starting from an empty one (only local moves of type enlargement), followed by a relative stagnation phase with small increases or decreases in quality which correspond to successive local moves of type removal-enlargement.

When the algorithm stops, the user can visualize the N best plans produced by looking at each of them in three windows. As the previous ones, the first two windows show all the non empty observation activities and the completion percentage of each observation. The third window details information about each observation, by giving its completion quality, its realization quality, and its resulting aggregated quality (see Figure 6). In the first window, the user can zoom on any part of the plan (see Figure 7 where revolutions are separated by thin vertical lines and where we can observe no-splitting observations, such as Observation 12, and periodic observations such as Observation 21 of periodicity 1 and tolerance 1 and Observation 23 of periodicity 2 and tolerance 1, all properly scheduled).

On the instance described in Section 3.5, the algorithm takes in general only some minutes to get plans whose quality is close to 0.97 or 0.98 *i.e.*, very close to 1 which is an upper bound on the plan quality. That means that, in the worst case, the best quality obtained is only 2 or 3% below the optimal one. Moreover, the time that the algorithm takes to get this result (some minutes) must be compared with the days of manual work that the *ISOC* should spend until now to get results of similar quality.

The integration of the *AIMS* tool in the chain of software components dedicated to the scientific operations of the *INTEGRAL* mission and its operational evaluation are currently in progress at *ESAC* in Madrid.

On the other hand, the *AIMS* tool has been successfully reused by the team of the *Politecnico di Milano* to deal with the third *APSI* case study: long-term planning for the *XMM-Newton* satellite (see (Lavagna and Castellini 2009)). Although *XMM-Newton* planning constraints and criterion significantly differ from the *INTEGRAL* ones, changes in the model and in the algorithm required only some weeks of work.

6. Discussion

In parallel with the development of the *AIMS* tool, we experimented the use the commercial *OPL* tool (http://www.ilog.com/products/oplstudio/) on the same problem. *OPL* is a high level modeling language for constrained optimization problems. To solve them, it uses either *CPLEX* which is able to handle linear and integer linear programming problems (with linear constraints and criterion), or *CP Optimizer* which is able to handle discrete constrained optimization problems (with constraints and criterion which may be not linear, but domains which must be finite). *CP Optimizer* has been recently extended to better handle resource constrained scheduling problems. It allows especially *interval* variables and temporal constraints on them to be defined. It allows for example a *noOverlap* constraint on a set I of intervals to be defined. This constraint enforces that all the intervals in I be disjoint. We wanted to evaluate the efficiency of the scheduling functionalities introduced in *CP Optimizer* on a real world resource constrained scheduling problem. In terms of modeling, we managed to model the *INTEGRAL* long-term planning problem as described in Section 3. In terms of solving, one of the main advantages of using generic constrained optimization tools such as *CP Optimizer* is there is almost nothing to do: only if necessary, to set as well as possible the few search parameters to which the user can access, such as for example the order in which variables are assigned and values are chosen.

Unfortunately, we did not manage to get with *CP Optimizer* plans of the same quality as the plans we managed to get with *AIMS*. For example, on the instance described in Section 3.5, *CP Optimizer* takes some tens of minutes to get plans whose quality is close to 0.90 and remains at this level after several hours of computing. We think that this is not due to some inefficiency of the algorithms associated with *CP Optimizer*. It is established that they are efficient enough to solve many real world optimization problems. This may be partially due to the huge size of the instance to be solved: some thousands of variables with some hundreds of possible values for each of them. However, we think that the main reason for such results is the difference between the search strategies used in *CP Optimizer* and *AIMS*. *CP Optimizer* uses an iterated stochastic greedy search enhanced by constraint propagation, where each greedy search is followed by a limited number of backtracks and a restart from an empty assignment with a increased number of allowed backtracks. On the other hand, *AIMS* uses an iterated hill-climbing stochastic local search in the set of consistent plans. We conjecture that, at least on the class of scheduling problems we considered, a local search which tries to iteratively improve a solution is far more efficient than a greedy search which tries to iteratively produce a better solution from scratch.

To be fair, we must stress that *OPL* is a generic tool whereas *AIMS* is a specific tool which has been especially tuned for the *INTEGRAL* long-term planning problem. If a new constraint should be introduced or if the definition of a constraint or of the criterion should be modified, with *OPL*, we would have only to modify the model whereas, with *AIMS*, we would have to modify the algorithm itself. As usual, efficiency must be paid and there is no free lunch!

Another sensible option would have been to use the free *COMET* tool (http://www.comet-online.org/). As *OPL* does, *COMET* offers a high level modeling language for discrete constrained optimization problems. Whereas *CP Optimizer* is black box on which the user has a strictly limited influence, *COMET* allows the user to design and to tune its own local search algorithm. However, *COMET* does not offer the scheduling functionalities that *CP Optimizer* and the core *APSI* framework both offer.

Finally, the main lessons we draw from this work are the following ones:

- it is possible to build an unambiguous mathematical model of the *INTEGRAL* long-term planning problem, including the definition of the constraints and of the criterion to be optimized;
- the core *APSI* framework is able to manage efficiently the basic scheduling constraints, such as the constraints of no overlapping between observation activities within each revolution;
- it is possible to build on top of the core *APSI* framework a tool able to guarantee the satisfaction of specific constraints and the optimization of a complex criterion;
- a tuned hill-climbing stochastic local search algorithm which combines the best ideas present in state-of-the-art local search algorithms can produce very good results in terms of plan quality and computing time;
- what previously required some days of manual work requires now only a few minutes of computing.

Acknowledgements

We would like to thank Alessandro Donati and Nicola Policella from *ESOC* for their support in the *APSI* study and all the members of the *APSI* team for fruitful exchanges. Special thanks to Amedeo Cesta and Simone Fratini from ISTC-CNR who developed the core *APSI* framework and helped us to use it properly. Finally, many thanks to Erik Kuulkers, Peter Kretschmar, and Delphine Anger from the *ISOC* at *ESAC* who helped us to understand and to model the *INTEGRAL* long-term planning problem and supported our work with enthusiasm.

References

[Aarts and Lenstra 1997] Aarts, E., and Lenstra, J., eds. 1997. *Local Search in Combinatorial Optimization*. John Wiley & Sons.

[Cesta and Fratini 2008] Cesta, A., and Fratini, S. 2008. The Timeline Representation Framework as a Planning and Scheduling Software Development Environment. In *Proceedings of the 27th PlanSIG-08 Workshop*.

[Kramer, Barbulescu, and Smith 2007] Kramer, L.; Barbulescu, L.; and Smith, S. 2007. Understanding Performance Tradeoffs in Algorithms for Solving Oversubscribed Scheduling. In *Proc. of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, 1019–1024.

[Lavagna and Castellini 2009] Lavagna, M., and Castellini, F. 2009. Advanced Planning and Scheduling Initiative's XMAS tool: AI for automating scheduling of XMM-Newton long term plan. Submitted to 6th International Workshop on Planning and Scheduling for Space (IWPSS-09).

[Smith 2004] Smith, D. 2004. Choosing Objectives in Over-Subscription Planning. In *Proc. of the 14th International Conference on Artificial Intelligence Planning and Scheduling (ICAPS-04)*, 393–401.

[Steel et al. 2009] Steel, R.; Niezette, M.; Cesta, A.; Oddi, A.; Fratini, S.; Cortellessa, G.; Verfaillie, G.; Pralet, C.; Lavagna, M.; Brambilla, A.; Castellini, F.; Donati, A.; and

Policella, N. 2009. Advanced Planning and Scheduling Initiative: MrSpock AIMS for XMAS in the Space Domain. Submitted to 6th International Workshop on Planning and Scheduling for Space (IWPSS-09).