Space Operations as a Guide for a Real-World Scheduling Competition

Eduardo Romero and **Marcelo Oglietti**
CONAE - Argentine National Space Agency
Paseo Colón 751, (1063) Buenos Aires
Argentina
{eromero, marcelo.oglietti}@conae.gov.ar

Abstract

The ultimate objective of the scheduling competition is to drive us developing frameworks that tackle real-life problems better. Whatever form we gave to this competition, benchmark problems will be the essential tools for identifying the best algorithms and frameworks. So, the future of the competition will be greatly influenced by the problems we choose. Oversimplified ones will produce a distortion from the competition's final objective. In this paper, we present three real-life scheduling problems from space mission operations. We extracted their core features to build a rich benchmark scheduling problem to be used in the competition.

Introduction

In view of a Scheduling competition, several issues need to be debated. Two important and related issues that must be addressed are: First, what features should a scheduling system have; and second, what encompass the core characteristics of what we call scheduling. In this article we try to answer these issues by means of describing three real-world problems emerged naturally from the normal operations of satellite missions.

Planning and scheduling space operations are critical tasks that consume many resources not only directly, but also indirectly. A suboptimal scheduling implies the under-exploitation of many resources of a space mission and, as a result, a significant increase of the final cost of the obtained products.

The need of planning and scheduling tools for space mission is a well-known fact. Accordingly, there is a lot of literature about the successful use of planning and scheduling techniques in these domains and since 1997, and every two years, it takes place the International Workshop on Planning and Scheduling for Space. Two recent examples are the scheduling of the Hubble Space Telescope activities (Ferdous & Giuliano 2006); and scheduling the services of ESA ground stations network (Damiani *et al.* 2006).

This shows an important role played by scheduling tools in cost effective space missions, and how the richness of space-mission operation scenarios makes them perfect candidates to be benchmark problems for a scheduling competition.

The scheduling problems we present here share some core characteristics that differentiate them from classical scheduling approaches. These differences can be summarized as follows:

- If we draw a parallel between the problems presented here and classical scheduling problems, we see that our problems are in some sense more constrained: we only have a few, highly tested and documented ways of carry out the tasks. This turns the temporal networks associated with our problems locally very constrained.

- Since preference and priority levels are spread all over our examples, we need suitable quality measures for the solutions. In classical scheduling we must optimize the used time to process a set of jobs or tasks. In our problems we also have to optimize over a family of quality and priority measures that increase the complexity of the problems.

- We need the modeling of a special kind of unary resources that might be configured in different ways. In all the problems we present here, the resources are basically unary. Noteworthy, the use of these resources implies the set of different parameters for their configurations. This information must be included not only because it is necessary for operations, but because it is used for constraint propagation. For example, in order to know whether two units can work in parallel or not according their configurations. Besides, some units can be configured in different ways for the same task. The actually used configuration for a task depends on the configuration of the other units involved.

- Changes in the schedule are continuously requested in the problems presented here, sometimes with no regularity, thus, we have to deal constantly with re-scheduling and a variable scheduling horizon. Besides, when a schedule has been set we send a notification of this to our clients, and for this reason re-schedule has a cost. Therefore, when re-scheduling, we have to modify as few as possible the previous schedule.

In the next section we describe the context in which these problems arise.

Scheduling in the Ground Segment of Satellite Missions

A space agency typically posses a net of ground stations and many computers and software systems destined to provide the ground support to its space missions. All these resources are usually grouped under the name of *Ground Segment*. We will refer here only to low-orbit satellite space missions, a very extended type of mission.

Once the satellite has been put in orbit, the duty of the ground segment is to ensure its good health and its profitable use. Due to the characteristics of a low orbit, the satellite is visible by a particular ground station only a few times per day, for around ten minutes by contact. These periods are usually called *visibility time windows* or, more informally, *passes*.

During a pass, the ground station sets a specific configuration of part of its equipment in order to communicate with the satellite by means of its antennas and other radio frequency equipment.

The communication between the satellite and the ground station can be unidirectional or bidirectional. For example, it is unidirectional when the ground station only downloads the science data coming from the satellite; and it is bidirectional when the ground station also sends commands to the satellite.

Hence, we have to schedule what passes are attended and the equipment used to attend them, taking into account that the equipment involved in a pass depends directly on the type of contact.

Also, we have to schedule the use of the satellite payloads, i.e. what we want the instruments on board the satellite do. The satellite users want to obtain data from the satellite, and the satellite activities must be scheduled considering these requests; but always taking into account the visibility time windows and many constraints that guarantee the satellite's good health. With all this information we have to decide what commands are going to be sent during the next passes.

The science data downloaded from a satellite is ingested and stored. It is raw data that needs to be processed to obtain the final products. This process sometimes needs to be performed in several steps, using in each step different software that could be available in some workstations only. Besides, the products have different priority levels that should be considered to set the order of production. For example, if a natural emergency makes indispensable certain satellite images.

Therefore, if we do a functional description of the necessary tasks to support a satellite mission, we identify three main groups. At CONAE they are organized as distinct services:

1. CONAE Ground Station Service (CGSS), responsible for all ground stations and in charge of managing all attended passes;
2. CONAE Mission Operation Center Service (CMOCS), responsible for the health caring and commanding of CONAE's satellites and instruments;
3. CONAE Users Ground Segment Service (CUGSS), in charge of providing the users with all the space information that CONAE can generate, independently of being its source CONAE instruments, other agencies instruments or mixed.

Usually, these services are oversubscribed and, for this reason, each service has a planning center where scheduling tools are vital for all of them. The scheduling problems are:

1. To determine which passes will be attended, together with the configuration of the ground station equipment that must be used for each one of them;
2. To determine which activities will be carried on with the satellite and its instruments, together with the commands that will be uploaded to them, the contacts with the ground stations network that will be requested for this, and a schedule of science data downloading;
3. To determine which users' requests of science data products will be answered, together with the contacts with the ground stations network that will be requested for downloading raw science data, and a schedule for the processing of raw science data in order to generate the final science data products.

Most of the dialog between the services, both between humans and computer systems, comes from the need of coordination between these three scheduling centers.

In the next three sections we give a simplified but realistic description of these services and their scheduling problems.

CONAE Ground Stations Service (CGSS)

The satellite Mission Operation Centers (MOCs) and the CUGSS request ground services of different types to the ground stations network. We have to make a schedule of the services that will be provided and the equipment that will be used for this.

There are many different services that can be provided by a ground station during a pass. But during normal operations the most common ones are: download of science Data Services (DS), and Telemetry, Tracking & Command Services (TT&CS). The DS consist in downloading science data from the satellite. The TT&CS consist in tracking the position of the satellite, determining the velocity vector and orbit data of the satellite; downloading the telemetry (real time and/or stored); and transmitting the commands to the satellite. The TT&CS are the most important services in the missions since they are provided in order to maintain the satellite in operations.

Although there are many technological alternatives in order to provide these services, e.g. the use of a specific bandwidth for transmission; there are some established standards. Specifically, in terms of the radio frequency technology, a ground station usually must be capable of:

1. Receiving an X-band signal from a satellite, usually used to download science data at high bit rates;
2. Receiving an S-band signal from a satellite, usually used to download telemetry;
3. Receiving and transmitting S-band signal to a satellite for commanding, usually used to tracking, real time telemetry, and command.

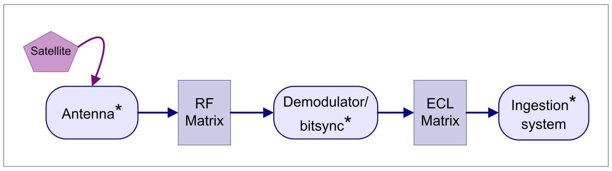

Figure 1: Science Data Acquisition

There are also many exceptions, e.g. stored telemetry is sometimes downloaded on X-band signal.

Science Data Download Services Science data Download Services (DS) consist in receiving the signal transporting science data from the satellite and, after some conversions, delivering the raw data. An extremely simplified sketch of the equipment involved in the process is shown in figure 1. In some cases we have in the ground station several units that can be used for the same specific task (e.g., we have several antennas); which sometimes are redundantly used. In the figure these equipment is indicated with asterisks.

The set of units as appear in the figure, i.e. {antennas, demodulators, ingestion systems} is the equipment used to acquire the data from a satellite, processing the radio-frequency signal coming from the satellite until the file with the science data is obtained. These units together with the way they must be configured and connected, is called *macro configuration*.

Typically, each satellite admits several macro configurations for a given pass and part of the scheduling problem is to choose a proper one. A macro configuration may be valued considering reliability and efficiency issues, since any satellite acquisition is intended to be done with the maximum level of redundancy in the used equipment.

In turn, although each satellite admits various macro configurations for a pass, not every unit can be used for every satellite. Due to compatibility and satellite specifications, some units cannot work together to attend a specific satellite. This is a common feature appearing in real-world problems remarkably different from some classic scheduling problems, where the usual assumption is that any machine can process any job.

Another important aspect for scheduling is that each piece of equipment or software system (from now on, both will be referred as units) must be specially configured with different settings for each macro configuration, and the change of this settings within macro configurations requires specific times. In other words, for scheduling activities, not only the visibility window must be considered, we have to look at the time needed to change the configuration of a unit. For example, a minimum time of 30 seconds is needed between any two passes with DS, because the antenna has to be placed in the correct position. This is a clear example of a time constraint in this scenario.

Telemetry Tracking and Command Services (TT&CS) This kind of service is very different from the previous one.

In DS we only receive a satellite signal that is usually an X-band signal, in TT&CS we have to be able to receive S-band signal and also, and really important, we have to be able to transmit S-band signal. The transmitted S-band signal contains the commands and, less frequently, software patches for the satellite payloads.

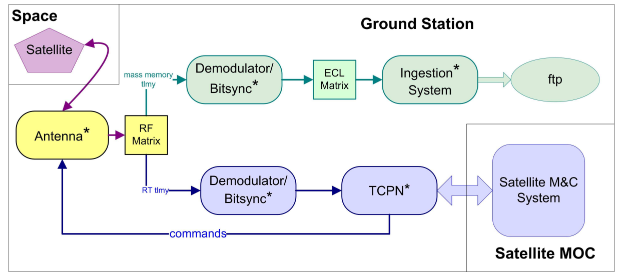

Figure 2: Telemetry, Tracking and Command

It is worth saying that to decide what commands have to be sent is a task of the mission operation center of the corresponding satellite, rather than a task of CGSS.

Besides, because of the fundamental role played by TT&CS in a satellite mission, we have requirements that specify a high redundancy in the equipment used for this kind of service.

About the technological aspects, the difference between this kind of services and DS is the equipment involved, e.g. for a TT&C service the used antennas must be capable of transmitting S-band signal.

Figure 2 shows a simplified sketch of the used equipment and how a TT&CS and an S-band signal download is carried out. Again, the asterisks indicate the cases where we have several units that can be used for the same specific task. These units are sometimes redundantly used.

Scheduling Satellite Contacts at the CGSS

Here we describe in more detail the scheduling of the contacts at the CGSS.

At CGSS we constantly receive requests of contact with satellites to provide DS and/or TT&C services. The requests of contact come from different sources: the satellite MOCs and the CUGSS requesting that some satellites must be attended. Each request of contact has associated a set of macro configurations that can be used to attend it. As we already said, a macro configuration is a list of the units and equipment used to provide a service, together with the units settings (i.e. the configurations of the units). A macro configuration may work for satellites with similar RF technologies, for example, the principal macro configuration for AQUA and TERRA satellites is the same. Table 1 shows two examples of macro configurations.

A schedule is just a set of macro configurations, including the initial and final times of each macro configuration, which are related to the times, orientation and other data of the corresponding passes. The beginning and ending time points of the macros and the orientation of the antenna are some of the settings that vary depending on the pass characteristics. But most of the other aspects of a macro configuration depend only on the satellite's technology and RF technologies

SERVICES	SAC-C X&S-Band	AQUA/TERRA
MACRO ID	108	418
ANTENNAS	7.3Ant & 13Ant.	7.3Ant.
DEMODULATORS	DATRON-7m & MICRODYNE	ALCATEL-7m
INGESTION SYS.	MEOS(Fep2,Ch1) & VexOri200 & Ing. S SACC	MEOS (Fep1,Ch1)

Table 1: Two Examples of Macro Configurations

and require little unfrequent changes. The principal aspects to consider when scheduling are the constraints between the use of the units. Most units can be viewed as unary, but configurable, resources. Some of the conflicts that have to be avoided or solved are the following.

1. Two macro configurations cannot share some of the units. For example, the antennas, the demodulators and bitesynchronizers, some of the ingestion systems that have not many instances, etc.
2. Two consecutive passes separated less than 3 minutes with macro configurations using the same antenna with different orientations cannot be both attended.

The schedule for each day is done with a week in advance, and frozen for possible changes a day before its execution, when it sent for its execution in the operations room.

CONAE Mission Operation Center Services (CMOCS)

A satellite Mission Operation Center (MOC) consists in the people, hardware, software and infrastructure needed to Monitor and Control (M&C) a satellite from earth. The CMOCS subsystem provides this service for all CONAE satellite services.

The satellite has on board the equipment needed to meet the mission requirements (for example, the satellite should be able of capturing images and/or measuring the earth magnetic field, etc). The on-board equipment can be classified according to its functions. A set of integrated pieces of hardware is called a *payload* when its purpose is to obtain certain data, usually for scientific uses. A set of integrated pieces of hardware is called a *subsystem* if its main functions are related to providing services to the payloads, including fundamental environmental cares such us attitude and temperature control.

The platform (all the subsystems together) has to maintain the satellite in orbit, has to provide services to each payload, and has to record its internal state periodically (for future analysis).For example, there are some basic subsystems present in all CONAE low-orbit satellites: Command and Data Handling (CDH), Power, Attitude Control System (ACS) and Mass Memory (MM).

Scheduling Satellite Activities at a Satellite MOC

Scheduling in a satellite MOC is the most complex problem of the three we are presenting. This is because, unlike the CGSS in which almost all resources are unary (not shared); in a satellite MOC we have several renewable and consumable resources that are shared by the different processes that have to be scheduled. Two important examples are: the use of the power on board the satellite, which availability depends on the eclipse times and the current charge of the batteries (a renewable resource); and also the use of the positions in the Time Tagged Commands (TTCmds) buffer (an amazing example of a consumable resource).

Scheduling with consumable and renewable resources increase the complexity. Nevertheless, as a first approach, we might not include these features into the problem and leave them for later constraint propagation, i.e. making a schedule and then propagate consumable resources constraint (e.g. checking if the consumed power is between accepted values). If the schedule violates some constraint it is modified until it is consistent. Although this division leads to non-complete scheduling procedures, this usually would not give troubles. This is because experience shows that the renewable resource constraints are almost never violated because of the existence of more restrictive constrains on the use of the systems and units on board the satellite. In what follows we describe the scheduling process without considering the renewable resources.

At each satellite MOC, we receive requests for activities or actions that the payloads and subsystem teams want to execute. For example, in the case of the payloads, to acquire some image, to reset some instrument or to upload some software patch. These requests can also be preventive or palliative measures requested by the flight operations team; such as a change in the parameters of the batteries or an orbital manoeuver.

The satellite MOC receives the requests together with orbit information. All this and a fixed scheduling horizon (that usually is of one or two weeks) is used to obtain the ways in which every request of activity might be attended.

For example, suppose a request corresponds to the acquisition of an image with a camera over some area. The possible ways to attend this request are established in the following way. First we specify a scheduling horizon and all passes over that area within the scheduling horizon. Next we filter the passes over that area without day light. For each of these selected passes we have a way to attend the request, namely, to turn on and turn off the camera in the specified times. We also have to establish when the acquired image is going to be downloaded. For each possible time of acquisition we choose any posterior pass over a ground station for downloading the acquired image. Hence, all we need to acquire the image is to know the particular units involved, their configuration, and the times of each configuration.

In this way, we see that some of the scheduling problems that have to be solved in a satellite MOC share some characteristics with the scheduling at the CGSS: we have some requests and partial configurations of the equipment to attend that requests. In this case scheduling is more complex because each action request might be composed of a few sub objectives separated in time. And this increase the number of possible ways to attend each action request. Following the parallel, we call macro configuration to each one of these

Sub Objectives	Unit: configuration(parameters)	Description	Execution Times
Camera Acquisition	Camera_HK_memory 1	Save house-keeping data of memory 1	1:30 min. before beginning of acquisition and until the acquisition ends.
	Camera_HK_memory 2	Save house-keeping data of memory 2	1:20 min. before beginning of acquisition and until the acquisition ends.
	Camera: high_resolution_stored_ acquisition(duration, size)	Begins acquisition	1 min. before start time of acquisition (due camera warmup) and until acquisition ends.
Dump of the Acquisition	Camera: CPM_counter(65535)	Raise parameter of particles counter up to 65535.	1 min. before beginning of dump and until 9 min after beginning of dump
	MassMemory: tx_x_on(mmrs1)	Turns on the x-band transmission	30 seg. before beginning of dump and
	Camera: mass_memory_dump(100000)	Begins the dump	Beginning of dump
	MassMemory: tx_x_off	Ends the transmission	8 min. after beginning of dump
	Camera: CPM_counter(10000)	Parameter of particles counter down to 1000.	9 min. after beginning of dump

Figure 3: The Two Parts of a Macro Configuration for a Camera Acquisition

sets of units, together with their configurations and times. For example, Figure 3 shows a macro configuration corresponding to a camera acquisition.

CONAE User Ground Segment Services (CUGSS)

In this dependency it is carried out the difficult task of interact with the users. At CUGSS we receive requests of satellite products by the users. The requests are mainly of two types: requests of products that are already in the products catalogue; or requests of satellite products which raw data has not been obtained yet, i.e. they are requests of acquisitions of science data.

Regarding these two types of requests, CUGSS has among others two separated functions.

1. CUGSS must filter and organize requests of acquisitions, making a previous feasibility analysis and assigning a priority level to each FR. CUGSS must also send the requests of raw data to CGSS or requests of acquisitions of science data to the satellite MOCs.
2. The CUGSS only stores the raw science data. This is because improved ways of processing raw data are obtained periodically and because the storing of all products would require unnecessary amounts of storage space. For these reasons the CUGSS only process raw data on demand, when a catalogue product is requested. Due to the great number of these requests, the use of the computational resources to process the raw data must be scheduled.

For space reasons, we will describe only the scheduling of the products processing, i.e. scheduling the use of computational resources in order to attend the requests of catalog products.

Products Processing Scheduling at CUGSS

As the previous examples presented in this paper, scheduling the products processing in the CUGSS can be formulated essentially in terms of requests and macro configurations to attend the requests.

For each product we have only a few configurations (or macro configurations) of the equipment that can process the corresponding raw data. Specifically, for processing a particular product we have to use some software systems that are generally installed in various computers. Furthermore, some computers have more than one software system installed that compete for the computer resources. In other words, we have an n-n relation between software precessing systems and computers.

Each system has several modes of operation to obtain different products. Even the same raw data can be processed in different ways by the same system, for example when different filters are applied to obtain images.

A few examples will be enough. Some of the satellites are processed to level 0 with the MEOS system, and its processing ends with the use of PGS system in another workstation. LANDSAT 7 is processed with ACS and LANDSAT 5 with PGS, both installed in the same workstation.

Besides, we have priorities within the requests that depend on the requesting users and the types of products. For example, when a natural emergency occurs, an image of the corresponding area is of high priority.

Each product requires a time of processing that can be accurately approximated.

We see then that this problem shares some fundamental characteristics with the previous ones. Namely, it can be stated in terms of a set of requests of different priorities and preferences; and these requests can be attended using some units specifically configured for that purpose.

A General Approach for Space Operations Scheduling Problems

In this section we identify the common characteristics of the three scenarios we have presented. We do this by means of a general description that includes the main features of the three scheduling problems presented above. This description also matches the architectural design of the CONAE Ground Station Services System described in (Oglietti 2006).

Action Requests In every one of the three problems we receive requests that have to be attended. These requests are of various types and comes with several instantiated parameters. We enclose these class of requests under the term *Action Request*.

At the CGSS we receive action requests of ground station services coming from the satellite MOCs or from CUGSS, i.e. requests for attending passes. We have one type of request for each satellite attended by the ground stations. In this case the parameters include the orientation, the beginning and duration, the type of service (DS or TT&CS), and some other data particular of that mission or pass.

At each satellite MOC we receive action requests from the payloads and flight operations teams that want to use the payloads and subsystems. We have one type of request for each action that a payload and subsystem can perform.

The parameters of the requests include the beginning and duration of the action, and other parameters that depend on the payload or subsystem. Also, the requests might require specific conditions to be executed, e.g. particular attitude or modes of operation.

At the CUGSS we receive action requests from the users that want certain products. Most of the products can be generated from raw science data already downloaded. We have a particular type of action request for each type of product. The parameters include the date of the data acquisition and other ancillary information, e.g. the convolution filter requested for image processing.

Units and Macro Configurations In every one of the problems we have several units that can be configured in different ways depending on the function they have to perform. These units are our basic bricks to construct macro configurations to attend an action request. In CGSS we have the antennas, the demodulators, etc. In each satellite MOC we have the payloads and the subsystems. In CUGSS we have several computers with different installed software systems.

Following the CGSS architectural design, the interface of each unit can be specified by a vector of variables of two types. A vector of read-write variables (`rw`) and a vector of read-only (`ro`) variables. The control of the unit is exclusively done through the setting of the values of the `rw`-variables. It is enough to read the values of all the variables of a unit to determine univocally the state of the unit. The values of the `ro`-variables offer all that can be observed about the unit change of state by exogenous or internal events (i.e. changes that happens independently of the values of the unit control read-write variables). This is necessary in order to model the non-deterministic behavior of the units (for example, the real angle of an antenna could be modified by strong winds and do not match exactly with the settle value, and therefore the sensed value should appear in some `ro`-variable value)

These variables can be of distinct basic types: integer or real numbers, strings, time points values, bytes, boolean, etc. or they can be restrictions of these basic types on some subset of values. Hence, in general, for each variable we consider that the corresponding set with the values that can be assigned to it is given. We name this set the domain of the variable.

This unified view of the units has several advantages. We can avoid the need of dealing with any particular characteristic of the internal behavior of the unit, and more important, we can know at any moment the state of the whole system by queering the values of all variables of all units. In other words, this architecture facilitates the monitor and control of the whole system.

In this paper we are mainly interested in the control aspects, used to set up the system to attend requests. The monitor aspects of the architecture are needed because, although we are not focusing on it here, the system is not deterministic and we have to be able to respond to any contingency, and more important, to see the incoming contingency.

Using the `rw`-variables a set of units can be configured in specific ways. We assume each unit has a stand-by configuration that is set by default if no other variables are passed to the unit. The instantiated variables of a unit together with the period of time that the a unit must remain with this configuration will be called here *partial unit configuration*.

To attend an action request we set up a set of units during the needed time period. In other words, we set up a set of partial unit configurations. We call *macro configurations* to each one of these sets of partial unit configurations. Each request has associated a set of macro configurations that can be used to attend it.

Actually, the macro configurations corresponding to each action request can be generated using the request type and the request parameters as the input. For example, the equipment used to attend a satellite pass only depends on the satellite, the type of service, and other parameters of the pass; and therefore, only the times of the macro configuration substantially vary between two passes of the same satellite.

Conflicts Between Macro Configurations There are cases where two macro configurations cannot be simultaneously set, for example when a unit cannot be shared; or when we need a minimum time separating two configurations (e.g., due to warm up times). In these cases we say that the macros have a *conflict*. For example, we have conflicts between a given unit and the same unit with a different configuration; and this a way of specify that a unit is a unary resource.

In what follows we briefly explain a way of describing conflicts in terms of a Disjunctive Temporal Problem (DTP) (Stergiou & Koubarakis 2000) formulation that helps to understand the problems.

We denote each unit with u_i. The units have associated different configurations that are defined through assignments of its `rw`-variables. We denote with $u_{iv} := u_i(\vec{v})$ the unit u_i configured with variables $v \in \mathcal{D}_{u_i}$, where $\mathcal{D}_{u_i}$ is the domain of the variables of u_i. We denote with U the set of all possible configurations of all units, i.e. $U := \{u_{iv}\}_{1\leqslant i\leqslant n, v\in\mathcal{D}_{u_i}}$,

We assign a time interval to each $u \in U$, representing the time that the unit must remain in the same configuration, and we call *unit basic configuration* to each one of these configurations. We denote this with the tuples (u, s, e), where s and e are the starting and ending time points of the unit configuration. Therefore, the set of all unit basic configurations is given by the set $\mathcal{UBC} := U \times \mathbb{R}^2_{x\leqslant y}$, where $\mathbb{R}^2_{x\leqslant y} := \{(x, y) \in \mathbb{R}^2 \,:\, x \leqslant y\}$.

We already said before that a macro configuration is a set of unit configurations. Hence, the set of all *macro configurations* is given by

$$\mathcal{MC} = 2^{\mathcal{UBC}} = 2^{U\times\mathbb{R}^2_{x\leqslant y}}.$$

We specify a *conflict* in the following way: $conf := (u_{i_1v_1}, c_1, c_2, u_{i_2v_2})$, where $(c_1, c_2) \in \mathbb{R}^2_{x\leqslant y}$. This means that if $u_{i_1v_1}$ is set in a time t' then $u_{i_2v_2}$ cannot be set in a time t'' such that $c_1 \leqslant t' - t'' \leqslant c_2$.

In symbols, it is equivalent to say that $(u_{i_1v_1}, t_1, t_2)$ and

$(u_{i_2 v_2}, t_3, t_4)$ are not conflictive with respect to $conf$ iff:

$$c_2 < t_3 - t_2 \ \vee \ c_1 < t_1 - t_4.$$

In this way, checking the consistency of a schedule (a set of macro configurations) consists in checking the consistency of a DTP. But these conflicts (the temporal constraints) are independent of the particular instance of the problem we are working on, because this specification of conflicts is independent of the requests.

A scheduling horizon is determined by two values, $h < H$, that specify the following constraints: $h < t_1$ and $t_2 < H$, for all basic unit configuration, (u_{iv}, t_1, t_2), present in any used macro.

Finally, we say that a macro configuration is conflictive with respect to a set of conflicts iff there exist two unit basic configurations in the macro configuration that are conflictive with respect to some of the conflicts.

Priority and Preference Measures When the action requests are received we have to choose, whenever possible, macro configurations to attend them. If we choose two macro configurations having conflicts, we have two possibilities. We have to use other macro configurations to avoid the conflicts or, in case this is not possible, we have to choose to not attend one of the action requests. For this reason we need some criteria to make these choices.

The preference measure of a macro configuration is easily determined considering its redundancy and reliability to attend the request, e.g. when we use two antennas instead of one to attend a satellite pass in the CGSS.

Any action request has associated a priority that depends on the request type and other particular characteristics of the request. These priorities are deduced from many factors such as previous agreements with other space agencies or the value of a certain satellite product. At CGSS, a request of TT&C service has a higher priority than a request of science data download service. In turn, a request of TT&C service coming from a mission in emergency has higher priority than other of TT&C in normal operations.

In the MOC an alarm detected in the telemetry values of a payload makes its Action Requests of the highest priority.

In the case of the UGSS, a request of an image of an area in emergency (e.g. with a volcano eruption) determines the priority of the request.

Among request of the same priority, preferences are used to choose what to do. Preferences are determined considering some minor factors. For example, at the CGSS, when we have to choose between two passes with the same priority, we usually attend the longest pass. Another preference criteria between two requests of the same priority is to attend the request that allows to attend the largest amount of requests of lower priority.

The main difference between priorities and preference is that priorities cannot be ignored, i.e. we cannot admit a schedule that attends a request of lower priority if, instead of it, we can attend a request of higher priority. Preference are used to solve conflicts between requests of the same priority.

Scheduling A solution consists in determining a set of configurations with no conflicts that respects the priorities and the stipulated scheduling horizon. Hence, a solution is a partial schedule of the units together with the parameters needed to configure each unit at each time, within the scheduling horizon. An aspect to be considered when scheduling is that, for high-priority requests, we give more importance to the robustness of the used configurations than to its time span.

Special Characteristics We want to highlight some special characteristics of these scheduling scenarios.

- We are constantly receiving new requests, and therefore we have to re-schedule periodically.
- Some constrains on the problem, e.g. each satellite's orbit parameters, are permanently refined periodically changing the current solution. From time to time the outcome of this is an invalid solution and the need for re-scheduling.
- When a schedule is frozen we send a notification of acceptance for each attended request. These notifications work like contracts, and hence, major changes in a schedule have a cost. Therefore, re-schedule has to be done minimizing the changes. This dynamic nature of these scheduling problems is not considered in classical scheduling frameworks. And hence, we believe that the capability of re-schedule meanwhile minimizing the changes should be evaluated in a scheduling competition.
- In classical scheduling problems we have tasks or jobs that can be processed with any machine. We are very restricted on how each request can be attended and this should guide the design of new approximation scheduling algorithms.
- We need to know the parameters to configure each unit because they are used for constraint propagation.

Related Work

We have already mentioned the existence of CSP-based AI planning and scheduling frameworks as (Laborie 2003; Cesta, Fratini, & Oddi 2004). Despite their high expressiveness, these frameworks cannot accurately represent the problems we have presented, mainly because of the nature of the units in our architectural design. In a representation that takes configured units as resources, it is difficult to express the temporal constraints in a succinct and flexible way. In turn, a representation that takes units as states variables will not take advantage of resource constraints propagation, an this is one of the most useful features of these frameworks.

We described the set of conflicts of our problems as disjunctive temporal constraints. For this kind of temporal problems there exist several algorithms that solve them efficiently, see for example (Tsamardinos & Pollack 2003). Nevertheless, these algorithms do not take advantage of the explicit and over constrained representation that macro configurations provide. The existence of macro configurations facilitates the search because they can be thought as small and local simple temporal problems (STP). This prunes considerably the search tree, since the search is made in the space of macro configurations and not in the space of all possible schedules.

The use of preference in DTPs is not novel, see for example (Peintner & Pollack 2004). The usual approach to assign preference in DTP is to assign a preference function to each time constraint. In this direction, we have identified in space-operations scheduling problems the need of a more general preference specification (the preference weights). These preferences are assigned to groups of constraints (the macro configurations), rather than to every isolated constraint.

There exists an approach to handle the complexity of pure configuration problems introduced in (Sabin & Freuder 1996). In this work, the construction of appropriate configurations is done by means of *Composite Constraint Satisfaction Problems*; a generalization of the CSP paradigm where variables can be instantiated in entire subproblems. These technics, adapted to scheduling frameworks, might be useful when working in the proposed scenarios.

Conclusions

In this paper we presented three particular scheduling problems from the field of space mission operations. These problems share the following core characteristics. We have:

- Action requests are of various types and have parameters;
- Configurable units and macro configurations, highlighting the importance of a hierarchical treatment;
- Compatibility constraints between the units' configurations, the constrains depending on the configurations;
- Conflicts between configurations;
- Variable scheduling horizons;
- Some problem constraints under permanent refinement;
- A continuous stream of action requests and the need of re-scheduling;
- Strong commitment with previous schedules;
- Several priority and preference measures;
- More attention to the robustness of the solution than to its time span.

In order to make the scheduling competition improve our stance in front of real-life problems, we need to include these characteristics in its benchmark scheduling problems.

We presented a way of describing constraints between units in terms of DTPs, showing the importance and usefulness of some modern planning and scheduling approaches when modeling real-world problems.

References

Cesta, A.; Fratini, S.; and Oddi, A. 2004. Planning with concurrency, time and resources: A CSP-Based approach. In Vlahavas, I., and Vrakas, D., eds., *Intelligent Techniques for Planning*. Idea Group Publishing. chapter 8, 259–295.

Damiani, S.; Dreihahn, H.; Noll, J.; Niézette, M.; and Calzolari, G. P. 2006. Automated Allocation of ESA Ground Station Network Services. In *Proceedings of the 5th International Workshop on Planning and Scheduling for Space (IWPSS-06)*, 71–80.

Ferdous, N., and Giuliano, M. 2006. Validating Resource Usage in Least Commitment Scheduling. In *Proceedings of the 5th International Workshop on Planning and Scheduling for Space (IWPSS-06)*, 82–91.

Laborie, P. 2003. Algorithms for propagating resource constraints in ai planning schedulin: Existing approaches and new results. *Artificial Intelligence* 143(2):151–188.

Oglietti, M. A. 2006. Domain Independent Planning for Space: Building a Bridge from Both Shores. In *Proceedings of the 5th International Workshop on Planning and Scheduling for Space (IWPSS-06)*, 316–325.

Peintner, B., and Pollack, M. E. 2004. Low-cost addition of preferences to dtps and tcsps. In *Proceedings of the 19th National Conference on Artificial Intelligence Proceedings*, 723–728.

Sabin, D., and Freuder, E. C. 1996. Configuration as composite constraint satisfaction. In Luger., ed., *Proceedings of the 1st Artificial Intelligence and Manufacturing Research Planning Workshop*, 153–161. AAAI Press.

Stergiou, K., and Koubarakis, M. 2000. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence* 120(1):81–117.

Tsamardinos, I., and Pollack, M. E. 2003. Efficient Solution Techniques for Disjuntive Temporal Reasoning Problems. *Artificial Intelligence* 151(1):43–89.