Astronomy and Computing 15 (2016) 90–104

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

A scheduling model for astronomy

M. Solar[a,*], P. Michelon[b], J. Avarias[c,d], M. Garces[d]

[a] Universidad Tecnica Federico Santa Maria (UTFSM), Av. Santa Maria 6400, Vitacura, Santiago, Chile
[b] Universite d'Avignon et des Pays de Vaucluse, France
[c] National Radio Astronomy Observatory (NRAO), Socorro, NM 87801, USA
[d] Atacama Large Millimeter/Submillimeter Array (ALMA), Alonso de Cordova 3107, Vitacura, Santiago, Chile

ARTICLE INFO

Article history:
Received 6 October 2015
Accepted 25 February 2016
Available online 15 March 2016

Keywords:
Astronomical scheduling problem
Dynamic scheduling
Astronomy scheduler

ABSTRACT

Astronomical scheduling problem has several external conditions that change dynamically at any time during observations, like weather condition (humidity, temperature, wind speed, opacity, etc.), and target visibility conditions (target over the horizon, Sun/Moon blocking the target). Therefore, a dynamic re-scheduling is needed. An astronomical project will be scheduled as one or more Scheduling Blocks (SBs) as an atomic unit of astronomical observations. We propose a mixed integer linear programming (MILP) solution to select the best SBs, favors SBs with high scientific values, and thus maximizing the quantity of completed observation projects. The data content of Atacama Large Millimeter/Submillimeter Array (ALMA) projects of cycle 0 and cycle 1 were analyzed, and a synthetic set of tests of the real instances was created. Two configurations, one of 5000 SBs in a 3 months season and another 10,000 SBs a 6 months season were created. These instances were evaluated with excellent results. Through the testing it is showed that the MILP proposal has optimal solutions.

1. Introduction

The telescope is the most important part of an observatory and astronomers want to use it as much as possible. An astronomical scheduler system acts as the organizer of the observatory. However, the main problem of an astronomical scheduler is not to manage observation requests, but also decides when each observation will be executed taking into consideration that use time is limited and the cost of observing time (specially in large astronomical projects) is expensive (Spotts, 2010).

The astronomical observation schedule process is a complex problem (Gómez de Castro and Yáñez, 2003). In first place, the scheduler has to deal with several constraints such as technical requirements, weather conditions, target pointing feasibilities, maintenance dates, telescope time availabilities, percentage of observational time of each associated institution, cost of opportunity between an observation and other ones ready to be executed in a specific time, etc.

Secondly, most of these constraints can change at any moment, making even more difficult to achieve a schedule of observations. Lastly, the astronomical observation scheduling problem is a multi-objective problem, several optimizations can be made to optimize different aspects of the observatory simultaneously.

Considering that most modern astronomical observatories already have some tools to perform a schedule process, there is still a huge part of human intervention (Mora and Solar, 2010). That means that is highly desirable to have a fully-automated dynamic scheduler to be able to react against any unexpected change of present conditions.

The astronomical observation scheduling problem faces several issues. One of them is that it is extremely relevant to optimize as much as possible the observatory resources, but also one of the most important aspects is that the scheduler must be able to adapt dynamically due changes in observing conditions.

Modern observatories like *Atacama Large Millimeter/Submilli meter Array* (ALMA) have a systematic methodology to achieve the process of scheduling observations. This process starts with a "Call For Proposals" in the early stage of an observation season in order to receive observational proposals (Nyman et al., 2010). Later, each of those pass through a revision process in which a specialized committee approves or rejects each proposal considering some aspects like: research impact of the proposal, technical requirements, telescope required time, nationality, etc., assigning an observation time for the approved ones. These approved proposals pass to a queue process in which the telescope

* Corresponding author. Tel.: +56 2 23037212; fax: +56 2 23037214.
E-mail address: mauricio.solar@usm.cl (M. Solar).

http://dx.doi.org/10.1016/j.ascom.2016.02.005

scheduler decides when will be proper to execute each observation considering their specific requirements, priority and current observing conditions.

This work presents the design, implementation details and results of a Mixed Integer Linear Programming (MILP) based scheduling algorithm to solve a simplified version of ALMA scheduling problem. We decided to use MILP because of its rigorousness, flexibility and extensive modeling capability (Floudas and Lin, 2005). Even when MILP has become one of the most widely explored methods for process scheduling problems, there is no documentation in the use of MILP to solve the astronomical scheduling problem.

The work aims the following:

1. Identify and study the main constraints necessary to develop a scheduling model for ALMA observatory.
2. Present a model of a scheduling algorithm using a MILP approach to solve the ALMA scheduling problem.
3. Implement a prototype of the proposed scheduling model.
4. Describe the instance data and generate several classes of instances under different scenarios that ALMA will face in the future.
5. Identify the main variables to analyze the performance of the proposed solution.

The present article is structured in five main parts. In Section 2, the state of the art is presented, where not only the main techniques already studied in the literature are described but also, the current status of how the most important astronomical observatories are facing their scheduling problems. Section 3 describes the theoretical model of the proposed solution and the global view of the algorithm. Section 4 shows the main statistics of real instances of ALMA, the testing methodology of the algorithm and the obtained results. Finally, Section 5 presents the main conclusions and describes some ideas to the future work.

2. Related work

A wide range of research related to the field of scheduling is possible to find in literature. Problems such as the "Job Shop Scheduling Problem" (JSSP) (Graham, 1966) are relevant in the field of dynamic scheduling because in most cases it is used as a base for constructing new scheduling algorithms. JSSP is considered to be a NP-Complete problem. It is based in trying to assign J_i tasks (each one with variable duration) over M_j machines in order to minimize the total makespan (total length of the schedule).

On the other hand, the "Flexible Job Shop Scheduling Problem" (FJSSP) is a natural extension of the JSSP. It allows that any task J_i can be assigned in any machine M_j without any particular consideration to minimize the makespan. This previous consideration also maximize the utilization of the machines allowing to have more free time to add more tasks.

Even though JSSP and FJSSP are theoretical models, several applications in the literature can be found using mainly Mixed Integer Linear Programming (MILP). In Floudas and Lin (2005) several formulations of short-term time scheduling models using a MILP representation are described and classified by their temporal representation in order to determine the most efficient way to produce a series of chemical products in a given time horizon. Some of the main constraint considered in all the representations are the limited production resources and that each product is created by a specific receipt and an established production chain. The facility can also produce different type of products and the resources to make chemical products are shared between all product chains. This research concludes that all MILP approaches solve efficiently the scheduling problem and also to continue researching in this topic using a reactive scheduling technique taking in consideration unexpected changes in a given period of the scheduling process by using stochastical variables.

Also, in Ierapetritou and Floudas (1998) another MILP-based scheduling model is described to solve a short-term time scheduling for continuous batch processes using an innovative approach. This study not only describes the model itself but also, two real applications based on consumption goods dispatching which their associated results are presented. The research concludes that the presented models obtain a higher performance in comparison to other models referenced in the research.

Other studies like Lee et al. (1996) and Chang et al. (2001) show other type of short-term time scheduling models using a MILP-based approach. The first study describes a model that solves an inventory management scheduling problem of a petrol industry. The production process line has two main parts: in the first process line, barrels are loaded into trucks and then, barrels are sent into several distiller machines. The study shows that a MILP-based approach using a discretization of time and supported by branch and bound technique is able to solve successfully real case scenarios. The second study describes another MILP-based scheduling model oriented to solve the hydro scheduling problem consisting on finding a proper schedule to fill several water tanks considering the customers demand and the current capacity of each water tank in each time. Also, this problem allows the cooperation of water tanks to supply the existing demand making the problem quite complex. The study describes a model formulation oriented to solve a simplified version of the problem for a real hidric central which solves efficiently the scheduling process using real-scale instances.

Castro and Grossmann (2006) show a short-term MILP scheduling model for a single layer batch process. The model tries to minimize the total cost of delayed processes supported by a time discretization by defining time slots in the schedule process. The study concludes that the use of time discretization improves noticeably the performance against other existing solutions specially when large periods of time are used.

Finally, it is important to mention that there are several heuristics and meta-heuristics, such as Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and others which are used to model and solve different applications of the scheduling problem. Wetter et al. (2015) show a greedy heuristic to construct an initial candidate solution to solve the scheduling astronomical observation providing a good compromise between time complexity and solution quality. The initial solution of this greedy algorithm is used as input to a simple local search technique (hill climbing) to construct a neighborhood of a given solution. Hill climbing (and any local search technique) is known to get trapped in local optima restricting the quality of solutions it may discover. Wetter et al. (2015) show an application of a SA as a way to overcome this short coming of local search algorithms. Next section explains the application of a GA in the CARMENES telescope (Garcia-Piquer et al., 2014), in SKA (Buchner, 2011), and an ACO algorithm used in CTA (Colome et al., 2012).

2.1. Scheduling in astronomy

Astronomy has had a constant interest in looking with more details the deep space in order to understand our origins. Hence, more observatories are being constructed with more advanced hardware and software technologies, more precise and sensible astronomical instruments like larger optical mirrors or more antennas. The construction of advanced infrastructure requires a huge amount of investment (in the order of billions of dollars) which can only be possible by a partnership between several institutions. The operative costs of a mega-scale observatory are quite high and so, is highly relevant to have some mechanisms to optimize the use of resources provided by observatories based on their demand. Consequently, many studies related to scheduling in astronomy have emerged these last decades.

In a general matter each astronomer applies for observing time in a telescope and the observatories manage these type of information to schedule those observing proposals for each season (Mora and Solar, 2010). Astronomical proposals include attached information that is relevant for the scheduling process. Most common information are observational requirements but in other cases, information like the association that each astronomer belongs to or the potential impact of each research are also relevant for the schedule process. In Bridger and Butler (2008) the *ALMA Project Data Model* (APDM, the document schema that ALMA uses to receive observation proposals) is described. When the observation proposals are sent by the astronomers, a Time Assignment Committee (TAC) evaluates those proposals being able to accept or reject according to several aspects like the observation feasibility or the scientific impact of the observation. For the accepted proposals a scientific priority score is used to represent the priority of every observation proposal to be executed into the telescope. In Hiriart (2010) the scientific priority score that ALMA uses to rank its projects is detailed. This ranking value is generated by a function that is defined by three numbers: (`score, rank, grade`). The `score` number is assigned by a specialized committee in an early stage of the scoring process. Later, the scored proposals which have been evaluated by different specialized committees are assembled in one ordered set. Since there is a possibility of collision of multiple projects with same `score`, the `rank` number is used to resolve these scoring collisions. Finally, `grade` number is used depending the project's type (*A, B, C* or *D*). Using the defined numbers, the scientific priority score S_i for project i is defined as Eq. (1) shows.

$$S_i = \frac{N_p - (\mathtt{rank}_i - 1)}{N_p} + K_i. \tag{1}$$

Where N_p is the total number of observational proposals and K_i according to Eq. (2).

$$K_i = \begin{cases} 4 & \text{if project is grade } A \\ 2 & \text{if project is grade } B \\ 1 & \text{if project is grade } C \\ 0 & \text{if project is grade } D. \end{cases} \tag{2}$$

Most observatories are composed by a joint-venture between several institutions. In this case, the assigned time to each proposal must fulfill the percentage of total time that each institution has right. Also, some observatories distinguish different methods to execute the observations. In the case of VLT, observations can be executed either "Visitor Mode" (requires the presence of the astronomer to execute the observation) or "Service Mode" (the observation is executed by an internal telescope operator). This last mode allows certain degree of flexibility to the observatory to handle an observation.

In Colome et al. (2012), the main components of an astronomical scheduler are described. Also a conceptual map of a generic control system of an astronomical observatory is described which reveals the role of an astronomical scheduler from the other subsystems.

The astronomical scheduling problem is an NP-Hard problem (Gómez de Castro and Yáñez, 2003). One of the main complexities of this problem are the unexpected changes of the variables involved in each schedule decision such as: weather conditions, telescope failures, unplanned maintenance, and others which make the problem hard to solve.

This problem can be treated as a multi-objective optimization problem: several optimizations can be made simultaneously in the observatory. In Gómez de Castro and Yáñez (2003), a scheduling model with the scientific priority score associated to the executed observations are maximized in a complete scheduling period. In Mora (2011) a different approach is described by a bi-objective model which both scientific priority score of the scheduled observations and the utilization time of the telescope are maximized. Finally, in Colome et al. (2012) and Granzer (2004) a list of relevant aspects of an observatory that can be optimized are discussed. The summarized list is as follows:

- Minimization of the inactivity periods of the telescope/observatory.
- Maximization of the assigned times to the scientific projects.
- Maximization of the scientific priority score of the executed projects.
- Maximization of the quality of the data obtained in the observations.

2.2. Scheduling in astronomical observatories

In this section a description of the most important astronomical observatories are presented with their current situation about how each handles at date their own scheduling process. Firstly, we will show the most important space-based telescope, then the ground-based telescope will be presented from optical telescopes to radioastronomy observatories.

2.2.1. Hubble Space Telescope (HST)

Hubble is a space-based telescope created by NASA and the European Aerospace Agency (ESA). Hubble is equipped with a 2.3 m telescope and several instruments to observe from near-UV to near-IR spectrum. The main purpose of this telescope is to explore the deep space taking advantage of not suffering problems produced by the atmosphere.

In Johnston and Miller (1994) *Spike* is introduced as a scheduling tool constructed initially to solve the Hubble's scheduling problem. Its design is quite flexible, able to be adapted easily to other scheduling problems (in fact, other astronomical projects already adopted *Spike*). It was developed using Lisp and its algorithm is based on a constraint satisfaction problem (CSP)-solver which considers both long-term and short-term scheduling phases. Several heuristics, such as conflict-minimization, reparation (using Artificial Neural Networks) and elimination (using Priority Lists) heuristics are used to support the scheduling process. All details of mathematical models and algorithms of *Spike* can be found in Johnston and Adorf (1992).

2.2.2. Chandra X-ray observatory

Chandra is nowadays one of the most important space-based telescope operated by NASA. It was launched to make X-ray based observations. The telescope follows a 64 h orbit around the Earth and is equipped with an ultrasensible instrument to capture weak X-ray signals which cannot be captured on Earth due atmospheric absorption. Chandra's telescope has several servos to adjust position over the current orbit and velocity.

In Bucher et al. (2008) a complete study of the main aspects of Chandra's scheduler are detailed starting with a complete description of how all the observation proposals are managed in the scheduling process and also the main factors that Chandra considers in its scheduler.

One of the particularities of Chandra scheduler is that several optimizations are considered, enumerated (by priority) as:

1. Minimize risk of the telescope
2. Maximize scientific objectives
3. Maximize observation time
4. Minimize the consumption energy of the telescope.

2.2.3. Joan Oró Robotic Telescope (TJO)

TJO is a small class observatory (0.8m telescope with a robotized dome) operated by Parc Astronomic Montsec Observatory located in the mountains of Montsec (1570 mosl) in Spain. One of the particularities of this small observatory is that its location is quite difficult to access, so the observatory is operated remotely. In that way, an effective scheduling system is crucial to optimize the use of the observatory avoiding any operational risk.

The TJO scheduling system considers a long, medium and short-term schedule process in which the best observations in the present period are considered to be observed during the respective night. The long-term schedule process observes the feasible observation proposals considering long-term attributes like visibility periods, planned maintenance periods and phases where some other object interferes the observation target. Later, a mid-term schedule process is used to select those observations that are feasible to be made during the night of the same day. Finally, the short-term schedule process uses short-term attributes, like weather conditions, to assign those observations to be executed in the telescope.

In Colome et al. (2010), a technical report describes the main aspects of the TJO scheduling system.

2.2.4. CARMENES instrument

Calar Alto high-Resolution search for M dwarfs with Exo-earths with Near-infrared and optical Echelle Spectrographs (CARMENES), a new generation 3.5 m optical telescope, will start its science survey in the year 2016. Located in Spain it is operated by a consortium of German and Spanish institutions.

CARMENES scheduler considers a short-term, mid-term and long-term scheduling phases and the scheduler is expected to handle thousands of observations to be scheduled in a whole observation season. The long-term and mid-term schedulers are based on Genetic Algorithms (GA), and the short-term scheduler is an astronomy-based heuristic in order to dynamic reschedule in less than 5 s.

CARMENES scheduler considers some simplifications such as: The number of targets to be observed are known, there are no dependencies between observations, and there is only a single telescope that is allowed to execute the observations. Despite, the scheduler considers some other factors that make the schedule process quite difficult to solve. One of this factor, the number of integrations, is dynamically calculated in the schedule process due to weather conditions are unknown.

Garcia-Piquer et al. (2014) describes in a general way the main components of the CARMENES telescope, including the scheduling system. This technical report describes how the scheduling system interacts with other subsystems of the observatory.

2.2.5. Green Bank Telescope (GBT)

The GBT is until now the largest single-dish ground based radio telescope and also the biggest movable structure in the world. It is located in West Virginia (USA) being part of the *National Radio Astronomy Observatory* (NRAO).

GBT adopted recently a prototype of an automatic dynamic scheduling system (formerly known as DSS) developed by NRAO. A complete description of DSS can be found in National Radio Astronomy Observatory (NRAO) (2013). The GBT "core-algorithm" is based on a rank-based solution to establish priorities between the observation proposals. In Balser et al. (2009) the ranking formula used by the algorithm to classify each observation to generate a priority list between all observations is described. The ranking formula used in the GBT algorithm including several variables such as: forecasted weather conditions, observation efficiency, target coordinates, and some other specific variables used for schedule observations are described.

Also, in Sessoms et al. (2009) a two-phase scheduling algorithm is described. In the first one, a Sudoku solver is used to find a solution on continuous-time observations. In the second phase, a Knapsack based algorithm is used to schedule into the rest of available time.

In McCarty et al. (2012) a functional prototype of the scheduling system is described. The prototype was implemented using Haskell. The source code of the implementation is given as part of several projects that NRAO has as an Open-Source Software (https://github.com/nrao).

2.2.6. Low Frequency Array Telescope (LOFAR)

LOFAR is the largest low-range radio telescope in the world and is composed by a distributed network of multipurpose low-cost sensors used mainly as an astronomical radio-telescope for 10 MHz–250 MHz range observations. LOFAR is also used for other fields such as geophysics or agronomy but in a less percentage.

One of the main features of LOFAR is the ability to execute multiple parallel observations at a time. For that reason, it is composed by several ground-based facilities that contain several antennas that can work in collaboration. The main one is located in Holland with 13 antennas and also, other 18 more facilities in the same country with less number of antennas. Other facilities are located in Germany, Sweden, France and UK, adding 31 more antennas.

At date, LOFAR does hundreds of observations weekly and it is expected that this number will grow in the following years, so the LOFAR scheduling system should react effectively at this scenario (De Jong, 2012).

In De Jong (2012) the main components of the current scheduling system of LOFAR are described. Scheduler system of LOFAR is defined as a tool that is used to support human-based decisions and also to solve execution conflicts during the execution of the observations. In fact, the operator is able to check several scheduling solutions, compare them, modify the scheduling solutions, or leave the scheduler to schedule by itself using a simulated annealing based algorithm. On the other part, the same paper describes the main restrictions of the LOFAR scheduling problem. The most relevant ones are: Current hardware status (current antenna operating frequency, current filters installed, data slots available, etc.), visibility of the observational target in the whole observation period, processing constraints and data-storage constraints.

Finally, in Grim et al. (2002) an implementation of an evolutive algorithm for LOFAR is discussed describing the main restrictions and variables for the simplified problem. The paper concludes that an evolutive approach for the LOFAR scheduling problem gets effective results.

2.2.7. Atacama Large Millimeter/Submillimeter Array (ALMA)

ALMA is at date the world's largest radio-astronomical telescope in the world and it is located at an altitude of 5000 mosl in the driest desert of the world. It was designed to observe in the millimetric range specter. During 2013 the observatory operated in Early Science mode (Cycle 0 and Cycle 1) and it was fully operative in 2015.

ALMA is one of the biggest efforts in the astronomical area ever made and was carried out by the partnership of the most important astronomical entities in the astronomical field such as: ESO, NRAO and NAOJ.

ALMA is composed by 66 antennas. These antennas can be grouped up to 6 sub-arrays which can perform observations independently, and also these sub-array configurations can be able to change its distribution during the observation season. Also, the antenna array is distributed in a physical extension and its baseline (the length of the farthest antennas of the array) can be up to 16 km.

Unlike optical observatories, ALMA can observe continuously 24 h a day.

ALMA platform is mainly based on the ALMA Common Software (ACS) which is a distributed instrumentation and control system that ALMA uses to construct most of their subsystems. These subsystems have a specific function inside the observatory operations. In particular, the scheduling subsystem receives the approved observations and dispatches the selected ones from the scheduler to the telescope control system. When the observation has been made, the corresponding data is delivered to the pipeline subsystem which is in charge of store and pre-process the data, and finally deliver a final product to the astronomer.

ALMA will exclusively operate in "Service Mode", so a real-time scheduler tool will be in charge to assign which observation will be executed considering the current restrictions.

In Mora and Solar (2010) a brief summary of the ALMA scheduling subsystem design is described. In first place, observation projects are divided in a minimal observation unit called "Scheduling Blocks" (SBs). Each SB has special attributes like execution time, execution feasibility in the observation season and also a dependency with other SBs. Also, observation proposals are stored (after the committee approval) into an initial queue maintained by a "Project Manager". A "Master Scheduler" (which has a constant interaction with the "Project Manager") coordinates the schedule of each sub-array in order to finally decides which observation will be made in each period of time in each sub-array.

In the vast literature about ALMA and its scheduling subsystem, some of the main researches are interesting to highlight:

- In Hoffstadt (2010) the design and implementation of a prototype of the planning mode simulator for ALMA is widely detailed. The main purpose of the planning mode simulator in the scheduling subsystem is to make scheduling simulations to study several scheduling algorithms under different simulated scenarios. Also, the projected scenarios of demand that ALMA expects to receive is around 18.000 SBs (peak), so ALMA's scheduler must be able to deal with that scale of demand.
- In Mora (2011) an ALMA scheduling model based on dynamic priorities is described. This model seeks to solve efficiently the ALMA scheduling problem taking in consideration that some dynamic constraints are present in the problem and affects the scheduling process. In it, some relevant assumptions are made highlighting two of them: The first one, the proposed scheduling model solves a simplified version of the ALMA scheduling problem in which only one array of all antennas is considered. Lastly, a time discretization is used assuming that each time slot has a duration of 1 h which is consistent with the maximum duration time of a SB. Finally, the work shows that the presented model was able to schedule a real set of observation projects of ALMA efficiently.
- In Colome et al. (2012) a survey of the main researches about astronomical scheduling is presented. In it, the ones listed concerning ALMA are algorithms based on Simulated Annealing (SA), Tabu Search (TS) and Evolutionary Algorithms (EA).
- Finally, in Clarke and Avarias (2012) the main variables considered for the dynamic scheduling problem of ALMA are presented and also the main requirements that ALMA needs for its scheduler.

2.2.8. Cherenkov telescope array (CTA)

CTA is a next generation ground-based very high energy gamma-ray instrument. The project is planned to start the operation phase in 2018. The CTA is expected to be the most powerful observatory for gamma-ray observations. This astronomical project is supported by more than 200 institutions and its construction cost is estimated to be around of €150 millions.

The observatory will be composed by 2 big arrays, each one located in each terrestrial hemisphere in order to have a full coverage of the whole observing horizon. The first array will be located in the southern hemisphere considering 60 antennas distributed over a surface area of 3 km^2, and the second one in the northern hemisphere considering 30 antennas distributed in a surface area of 1 km^2.

The preliminary design of the CTA Observatory is described in Actis et al. (2012). In it, ALMA Platform (ACS) is considered to be used as a base for most subsystems of CTA and so, the most relevant parts of ACS considered to be used in CTA platform are detailed. On the other hand, CTA considers ambicious requirements (e.g., dead time of the telescope less than 10 s) and also the observatory expects to provide several types of observation modes.

The scheduling problem that CTA is expecting to have is quite similar that ALMA's one. In fact, the main purpose of CTA is the assignment of multiple observations to multiple arrays that can be formed in order to maximize certain criteria like the scientific impact of the observations that have been executed.

In Colome et al. (2012) a summary of the major requirements of CTA are described. The preliminary design of the CTA scheduling system expects to have two phases of schedule (short-term and long-term) and also a dynamic algorithm with a fast response time if a re-schedule is needed. In the same document, a preliminary design of the dynamic algorithm based on Artificial Intelligence is detailed. In summary, the scheduler for CTA is based on an algorithm composed by a Guarded Discrete Stochastic Neural Network and using Constraint Satisfaction Problem (CSP) for constraint propagation. An Ant Colony Optimization (ACO) algorithm is also used to avoid hard constraint violations. One of the benefits of this algorithm is that scheduling conditions can be changed during execution without the need to start from scratch.

2.2.9. Square Kilometre Array (SKA)

The SKA, which Australia and South Africa were chosen as co-host, is aiming for scales where current approaches—in construction, operation but also scheduling are insufficient. Although manual scheduling is common today, the problem is becoming complicated by the demand for: (1) independent sub-arrays doing simultaneous observations, which requires the scheduler to plan parallel observations; and (2) dynamic re-scheduling on changed conditions. Both of these requirements apply to the SKA, especially in the construction phase. An automated long-term scheduling system is sought that allows parallel observations, is reactive to resource availability and operator preferences, can re-schedule once these change and, last but not least, is scalable. One interesting aspect of the SKA scheduler, as in ALMA and LOFAR, is to allow parallel execution of observations. This means that a job needs several machines at a time (e.g. any subset of a given size), while another uses another, distinct, subset. This formalism is incompatible with job-shop scheduling that requires a job to go through machines in order, but does not allow it to use multiple (possibly varying) machines sets. Accepted proposals are input to the Observation Design process, creating Scheduling Blocks (SBs). SBs allow the planning function to create long- and short-term queues, and the short-term queue is then carried out at the telescope, with SBs executed in the form of their science scripts, interacting with Telescope Management sub-system. There are distinct off-line tasks related to proposal preparation and submission, proposal review and long-term observation planning and preparation, as well as on-line functions that relate to short-term observation planning and execution, including handling special requirements like Target of Opportunity (ToO) observations. The Genetic Algorithm (GA) is well-known to perform poorly in order-based encodings. This is due to the fact that mutation operations are difficult to write so that the GA can efficiently sample the problem space. Hence the recommended time-indexed encoding has been used by Buchner (2011). In his work, Buchner (2011) found that simple algorithms, in particular just going through the schedule and allocating the highest-priority observation possible at each time slot, lead to very good first-cut solution in the test dataset looked at. The GA can select the best heuristic and thus keep the approach robust to changes in the characteristics of the problem. It may help to make minor improvements, but also allow to update the schedule on changed requirements.

2.2.10. Large Synoptic Survey Telescope (LSST)

The LSST science specifications require operating the telescope for a 10-year survey of the visible sky with frequent revisits in multiple filters. During regular night operation, an observatory scheduler will be used to select fields to be observed on a cadence of one every 30 s. To maximize the survey efficiency, real time adjustments to the observation schedule will be assessed using input parameters like cloud cover. During regular night operation, LSST will use an observatory scheduler based on the same principle, that will select fields to be observed in a specific filter for overall survey efficiency. One of the input parameters of the scheduler will be the cloud cover of the sky during the night. In astronomy, nights are usually classified in terms of photometric hours and image quality. The *photometricity* of the night is related to the meteorological cloud cover of the sky. Traditionally, observations are taken during photometric and spectroscopic hours. During spectroscopic nights, LSST observations will be carried out since it gathers temporal information, and because image quality can still be good. The scheduler will be able to adjust the sequence of observations based on the photometric conditions of the different areas on the sky in real time, which is a hard requirement for the scheduling application. The purpose is to maximize the efficiency of the data collection time and the efficiency of the survey mission. See detailed information in Sebag et al. (2007).

2.2.11. Thirty Meter Telescope (TMT)

According to the information provided in http://ast.noao.edu/system/us-tmt-liaison/survey-faq, each TMT partner will establish and operate its own observing proposal and time allocation process. Partner shares of observing time will be divided into equal amounts of bright, gray and dark time, evenly spread across each semester's available time. TMT will receive the information on time allocation from the partners and will produce a merged observing schedule for each semester.

TMT will initially support two main observing modes (http://www.tmt.org):

- PI-directed mode: Classically scheduled blocks of observing time (generally in units of full or half-nights, but in some circumstances shorter) will be assigned to PIs or to TMT Partners. The PIs are responsible for carrying out the observations. Remote observing will be supported.
- Service observing: TMT staff members execute shorter, pre-defined observations on behalf of PIs in an ordered sequence. PI eavesdropping will be supported.

There are also other requirements to special assignations:

- Queue scheduling: Initially, TMT will not use observatory-wide adaptive queue scheduling. However, tools will be provided to allow adaptive scheduling within a PI- or partner-directed block of observing time. Partners may then run their own queues. TMT could implement observatory-wide adaptive queue scheduling at a future time, depending on the wishes of the partner communities, but additional staffing would be required, beyond what is in the baseline science operations plan.

Project	ACO	Spike	Disp	SWO	DP	EA	TS	SA
HST		*						
VLT		*						
GEMINI			*					
TJO			*					
SOFIA				*				
GBT					*			
ALMA						*	*	*
LOFAR						*		*
CTA	*							
SKA						*		
CARMENES						*		

Fig. 1. Scheduling techniques in observatories.

- Cadence observing: Programs that require monitoring observations at particular times will be accommodated in the pre-planned service observing mode.
- Target of opportunity (ToO): There is a TMT policy for cross-partner ToO interrupts, with detailed accounting for how ToO observing time is charged to the partners. Typical response time from trigger notification to ToO observing, including telescope slew and acquisition time, is expected to be 10–15 min.

Like TMT there are currently two more Extreme Large Telescope (ELT) projects: the Giant Magellan Telescope (GMT) project (http://www.gmto.org) fully commissioned in 2024, and the European Extremely Large Telescope (E-ELT) project (www.eso.org/e-elt), which first light is targeted for 2024.

2.2.12. Summary of scheduling techniques

In Mora and Solar (2010) and Colome et al. (2012) it can be possible to find a summary of the most common scheduling techniques used in the most important astronomical observatories and also their interaction between other subsystems are described. In particular, in Colome et al. (2012) a comparative chart of the different scheduling techniques for several astronomical observatories is described to summarize the most relevant studies related to scheduling applied to astronomy (Fig. 1). In this comparative chart, the algorithms/solutions included among astronomical observatories are: *Spike*, Ant Colony Optimization (ACO), Dispatcher (Disp), Squeaky Wheel Optimization (SWO), Dynamic Programming (DP), Evolutionary Algorithm (EA), Tabu Search (TS) and Simulated Annealing (SA).

There are other studies of astronomical scheduling problem like the case of the Stratospheric Observatory For Infrared Astronomy (SOFIA) detailed in Civeit (2013), the Very Large Telescope (VLT) detailed in Johnston (1988), Colome et al. (2012), Silva (2002), Gemini observatory that consists on two identical telescope facilities, one located in northern hemisphere (Hawaii) and the second one in southern hemisphere (Chile) (a detailed description of the scheduling system can be found in Puxley (1997)), and so on.

Most astronomical observatories still have some scheduling phases in which human intervention is necessary. This reveals that there is a lot of open opportunities to still work in other approaches to solve this problem in order to have a fully automatized scheduling tool that optimizes efficiently the observation process of an observatory and responding efficiently to any unexpected change that can affect the scheduling process.

3. Solution model

The ALMA scheduling problem is a complex problem with a huge amount of variables to be considered. Below, some main variables that affect the ALMA scheduling problem are described while detailing the respective assumptions taken in account:

1. **Antenna-array configurations:** ALMA is composed by 66 movable antennas of different characteristics which can operate independently or in collaboration. In collaborative mode, each antenna can be moved from one place to another according to the observation requirements, and so each antenna-array can be more or less convenient for different observation necessities. In the hardest scenario, to check which antenna array is best suitable for a specific observation, a combination of $66! \approx 5 \times 10^{92}$ different antennas' configuration should be analyzed. Also, each sub-array can work independently making feasible to run in parallel multiple scheduling algorithms at a time making the problem more difficult than it already is. Avarias (2014) provides a scheduling algorithm for different array configurations within an observing season. In this study, only a unique all-antenna array (working together as a single telescope) is considered, so the multi-array scheduling problem will not be analyzed in this work (see Avarias (2014)).
2. **Unexpected unavailable periods:** Some unexpected maintenance periods can appear randomly during the operations of ALMA and also, other technical failures may force the shutdown of an antenna for a specific period of time. In this work, all the antennas are assumed to be fully available to make observations without any unexpected failure or unavailable periods of time during the full observational season.
3. **Observational season:** The observational season is considered to be a 6 months fully-available period.
4. **Time assignment percentages per institution:** The time assignment percentages are not taken in account. That means that all observations are treated equally without any distinction according to which institution each observational proposal belongs.
5. **Observational proposals:** Observational proposals are already known at the beginning of the scheduling process. This means that some relevant information like their scientific priority score, Scheduling Blocks and their availability to be executed in the whole season (using basic information like target availability in the sky) are also included.
6. **Scheduling Block (SB) duration:** Every SB has a variable duration. A SB is the most basic unit that can be executed in the telescope but it does not necessarily mean a unique observation. Some observations to a specific target need a huge amount of integration time and therefore, observations are divided in several SBs of a given amount of time each. This study considers that all SBs have a static duration of 30 min as an approximation based on the past experiences in ALMA's early science program (Cycle 0).
7. **Time discretization:** The scheduling model considers a time discretization with a Time Slot duration of one SB (30 min).
8. **Weather conditions:** Weather conditions are assumed to be already known in the short-term period (48 h).

3.1. Formalization of the problem

Considering the previous assumptions and considerations, the problem can be formalized. In first place, a time horizon H is defined to represent the total available time of the observation season of the observatory. A set of n observational projects $P = \{P_1, \ldots, P_n\}$ is defined to represent the accepted observation proposals that will be scheduled. Each observational project P_i is composed by a variable set of Scheduling Blocks $O_i = \{SB_{i1}, \ldots, SB_{ij}\}$. O_i must contain at least one SB and also, a scientific priority score S_i is defined as a value that represents the priority based on the scientific impact of the observation project.

A set of Time Slots $F_{ij} = \{TS_1, \ldots, TS_\ell\}$ tells when the SB_{ij} is feasible to be executed (considering only static variables) during the whole time horizon. Static variables are statically provided at the beginning of the period and do not vary during the execution. Some of these static variables are: the number of total projects to be scheduled, the number of SBs by project, the scientific priority score of each project, total execution time for each SB, total period time, long-term time slot duration, etc. Finally, some SBs may depend on other SBs of the same project. For this reason, a Scheduling Block SB_{ij} may define another Scheduling Block SB_{ij^*} to establish a precedence constraint which means that SB_{ij^*} must be executed before than SB_{ij} in the telescope.

3.2. Key components of the proposed solution

The proposed solution solves the astronomical scheduling problem inspired in the ALMA observatory considering the above described assumptions. The proposed algorithm collects the most important aspects of some other scheduling models applied in astronomical observatories proposing to establish a greedy-based heuristic to optimize as much as possible this complex problem. In particular, this algorithm is mostly based on the study made in Mora (2011) in which a dynamic scheduling model based on priority queues is described to solve the ALMA scheduling problem. In this work, a complete technique is proposed instead of using priority queue to force the algorithm to get the optimal solution in each short-term period of the schedule season.

Fig. 2 shows the ALMA dataflow, which describes the lifecycle of the observation data and also, how the scheduling subsystem is involved to decide which observation should be made in each lapse of time.

In the following sections, some main components of the scheduling solution are described.

3.2.1. Long-Term scheduling process

The proposed algorithm is based in a two-phase scheduling process because the astronomical scheduling problem can vary over time due dynamic variables. The first phase is a long-term schedule process in which SBs are arranged into long-term time intervals. These long-term time intervals are constructed by a discretization of the whole season time by trying to set a proper lapse of time in which SBs are feasible (considering only static variables) to be executed in the telescope. It is important to remark that every long-term time slot (LTS_i in Fig. 3) must be an exact multiple of the value of time of TS_i.

During the schedule process, whenever something changes during the scheduling process (i.e. SB was executed in the telescope, a project concludes or a project is unfeasible), the long-term schedule process adapts its schedule map. Fig. 3 shows the relation between LTS and TS, as an example LTS could be a 24 h window, and TS could be a 30 min slot. Each TS_k has an x indicating that the SB_ij is feasible to be executed in that TS_k.

3.2.2. Short-term scheduling process

The second phase in the proposed algorithm is a short-term scheduling process in which a definitive schedule is calculated for each period of time of the whole observing season. During a specific short-term period of time, feasible observations that were already arranged in the long-term phase are introduced to a new schedule phase in which more variables are considered (dynamic variables). This schedule phase can be treated as a Constraint Satisfaction Problem (CSP), so several techniques can be used. In this particular case, a complete heuristic is used to obtain the optimal value of the objective function used in the scheduling model.

Fig. 4 is an example of how a schedule for a specific lapse of time is constructed.

3.2.3. Dynamic parameters

One of the main aspects to handle every scheduling solution oriented to solve the astronomical scheduling problem is to

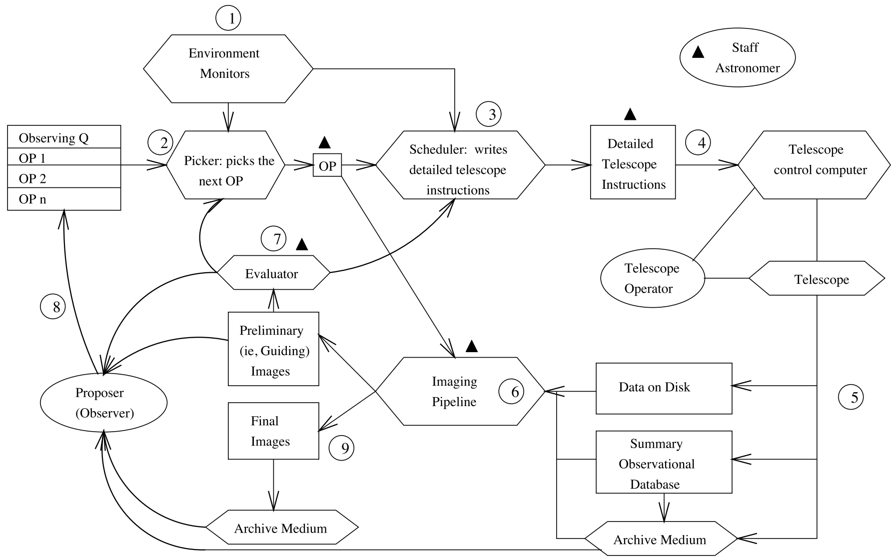


Fig. 2. ALMA's dataflow schema.

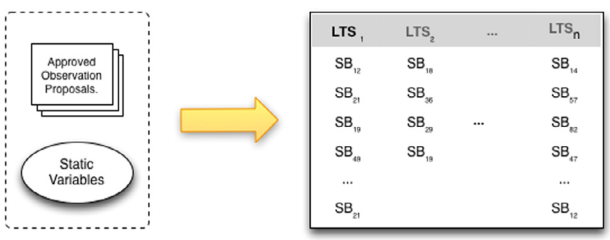


Fig. 3. Representation of a long-term schedule.

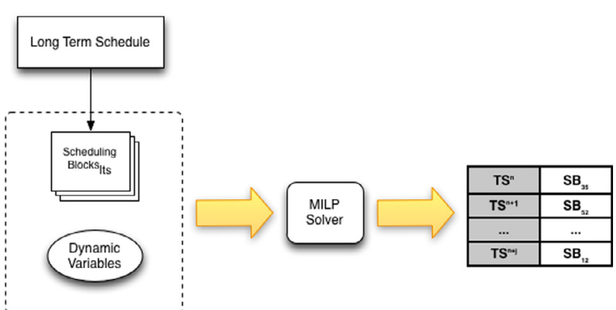


Fig. 4. Representation of a short-term schedule.

deal with dynamic constraints. Dynamic variables cause that the problem changes and so, a re-schedule is required to re-calculate a new proper schedule that considers the updated conditions. Some of the main dynamic parameters present in this problem are:

1. **Weather conditions:** Some observations require very specific ranges of certain weather variables such as: atmosphere vapor water percentages, temperature, atmosphere opacity, etc. Those data in general are obtained by an online-forecast.
2. **SB precedency:** SB precedency does not allow that any SB with precedence constraint can be executed if its predecessor is still pending. This condition is constantly updated whenever the predecessor is executed. Over the time, new opportunities can arise if any SB gets free of any precedence constraints.
3. **Project percentage completion:** Considering that the main idea is to conclude observation projects, it is important to know the actual status in terms of completion percentage of projects linked to the SBs that are taken in account for the present scheduling decision.
4. **Derived stochastic data:** Probabilities are also useful to take in account specially in dynamic problems. Information like the probability that a SB is feasible in the future is also relevant to take in account for the present decision.

3.3. Mathematical formulation

The scheduling process can be modeled as a CSP problem. The proposed mathematical formulation for this scheduling solution is described below.

3.3.1. Parameters and assumptions

1. Total scheduling time of short-term period (h) is discretized in equal-sized time slots. The size of each time slot must be as big as possible in order to wrap at least one SB (ℓ). Also $0 \leq \ell \leq h \leq H$.
2. The remaining percentage for the observational project i (P_i) to be completed is defined as ρ_i, defined according to Eq. (3). In mathematical terms, by considering $PendingSBs(P_i)$ as the sum of SBs of the project P_i that are still pending to be executed, and $totalSBs(P_i)$ as the sum of all SBs of the project P_i, then:

$$\rho_i = \frac{\#PendingSBs(P_i)}{totalSBs(P_i)}. \tag{3}$$

3. The future feasibility of a SB j that belongs to a project i is defined as φ_{ij} and means the probability in which the SB j will be available in the future. In other terms, by considering $feasibleSlots(SB_{ij}, t) = 1$ if SB_{ij} is feasible in the Long-Term Time Slot t or zero otherwise, then:

$$\varphi_{ij} = \frac{\sum_{t=\tau} feasibleSlots(SB_{ij})}{\#longTermTimeSlots_{t=\tau}(SB_{ij})}. \tag{4}$$

3.3.2. Variables

- $X_{ijt} = \begin{cases} 1, & \text{If } SB_j \text{ of the project } i \text{ is executed in the timeSlot } t \\ 0, & \text{Otherwise} \end{cases}$
- $Y_i = \begin{cases} 1, & \text{If the project } i \text{ has been completed} \\ 0, & \text{Otherwise.} \end{cases}$

It is important to mention that Y_i is treated as a soft variable in the CSP model. This means that Y_i will only be present in the model when a Project P_i is near to be finished (all their SBs are present in the current short-term period, so not necessarily already executed).

3.3.3. Model formulation

- **Objective functions:**
  1. Maximize the Scientific Throughput (ST) or the number of executed projects with higher scientific priority score S_i defined in Eq. (1) (Eq. (5)).

$$ST = \max \sum_i S_i \cdot Y_i. \tag{5}$$

  2. Maximize the effective telescope time utilization of each short-term period (Eq. (6)). In other way, this can also be understood as a minimization of the telescope idle time by choosing the best SBs configuration as soon possible but also taking in consideration that project completions are also desirable. This decision tradeoff is set up by benefit coefficients detailed in Eqs. (7) and (8).

$$\max \sum_i \mathcal{B}_i \cdot Y_i + \sum_i \sum_j \sum_{t=h_0}^{h_n} b_{ijt} \cdot X_{ijt} \tag{6}$$

where:

$$\mathcal{B}_i = \frac{S_i^2}{\rho_i \cdot totalSBs(P_i)} \tag{7}$$

$$b_{ijt} = \frac{S_i}{\rho_i \cdot \varphi_{ijt}}. \tag{8}$$

As an aside, it is important to clarify that the proposed algorithm optimizes directly Eq. (6) in each short-term period but Eq. (5) is not explicitly optimized.

- **Subject to:**
  1. Consider only feasible SBs for each project in each TimeSlot (Remember that F_{ij} is a set of Time Slots indicating when the SB_{ij} is feasible to be executed considering only static variables during the whole time horizon):

$$X_{ijt} = 0 \quad \forall t \notin F_{ij}. \tag{9}$$

  2. The telescope array can only execute one SB in a specific time:

$$\sum_{i=1}^{n} \sum_{j=1}^{n_i} X_{ijt} \leq 1 \quad \forall t \in \{1, \ldots, h\}. \tag{10}$$

  3. Precedence constraints between SBs of same project:

$$X_{ijt} \leq \sum_{\tau=1}^{t-1} X_{ij'\tau}$$

$$\forall i = 1, \ldots, n;\ \forall j = 1, \ldots, n_i;\ \forall t = 1, \ldots, h; \tag{11}$$

if j' is a predecessor of j.

  4. Each SB is executed at most once:

$$\sum_{t=1}^{h} X_{ijt} \leq 1 \quad \forall i, j. \tag{12}$$

4. Analysis and results

The implementation of the proposed solution described in Section 3 was used to analyze the performance of this algorithm by investigating the behavior of how different parameters affect important performance metrics such as the objective function value or execution time. The results of real scale instances are presented by detailing the obtained results of the most relevant performance metrics of the proposed algorithm.

Table 1
Main characteristics of ALMA cycle 0 data.

Number of projects	463
Total number of SBs	950
Number of SBs with precedence	0

Table 2
Main characteristics of ALMA cycle 1 data.

Number of projects	515
Total number of SBs	1379
Number of SBs with precedence	0

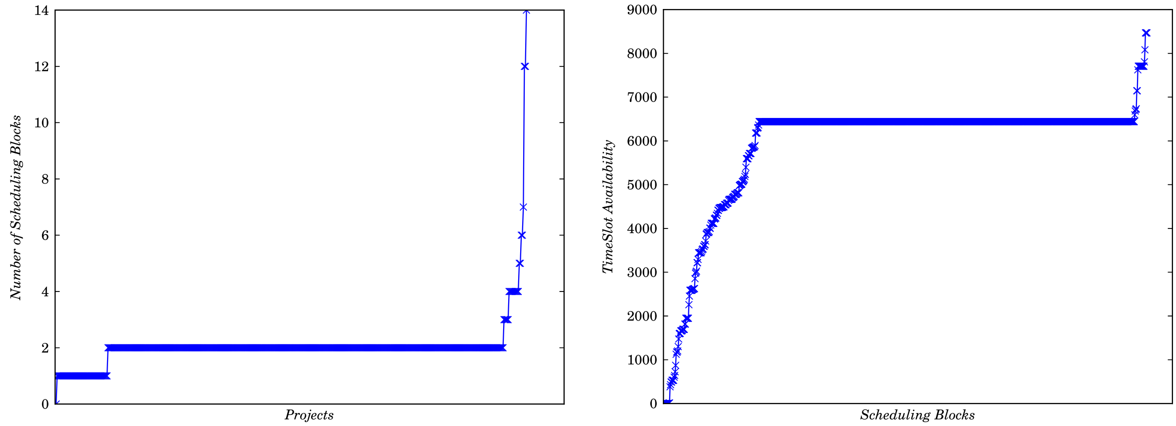
The implementation was developed using C++ compiled with the Clang 500.2.79 under the OSX 10.9 (Unix environment). The framework IBM ILOG (CPLEX, 2014; Watson and Cacioppi, 2014), a high performance multipurpose solver which use advanced techniques to solve MILP problems, was used to implement the MILP model described in Section 3.3.

4.1. Simulation environment

Considering that ALMA is still under development, there is still a lack of a consistent volume of real data to analyze the proposed algorithm. In this way, synthetical instances were generated looking forward to be as close as real as ALMA's scheduling data. It is important to clarify that ALMA is still in early science program and each operation phase are identified as "Cycle". At date, only two operation cycles have been completed (Cycle 0 and Cycle 1 of early science program). By using general characteristics of the scheduling data of each Cycle, real-scale and test-scale instances were generated to analyze the behavior of the algorithm. The characterization of each ALMA Cycle are detailed below:

- **ALMA Cycle 0 scheduling data:** ALMA's Cycle 0 call for proposals started on the first quarter of 2011 and operations began in September of 2011 with approximately 500 h of telescope time availability (without considering maintenance periods). The main characteristics about the volume of data of Cycle 0 ALMA operations are detailed in Table 1.
  Also, some main distributions are detailed in Fig. 5. Distribution (a) shows that in average 84% of the projects have two SBs, 11% of projects have one SB and the final 5% of projects have either 3, 4, 5, 7 or 14 SBs. Concerning distribution (b), it shows that 77, 6% of SBs have 6441 feasible Time Slots and the rest may have between 0 and 6441 Time Slots or 6441 and 8468 feasible Time Slots distributed in a pseudo-linear way.
- **ALMA Cycle 1 scheduling data:** ALMA's Cycle 1 call for proposals started on May 31st of 2012 and began operations in January of 2013 with a similar telescope availability time of Cycle 0 operations. Some of the main characteristics about the volume of data of ALMA's Cycle 1 operations are detailed in Table 2.
  The main distributions of ALMA's Cycle 1 scheduling data are detailed in Fig. 6. Distribution (a) shows that 34% of total projects have one SB, 22, 9% have two SBs, 13, 2% have three SBs, 14% have 4 SBs, 12, 6% have 5 SBs and the final 3, 3% have between 5 and 24 SBs. Distribution (b) shows that the Time Slots and SBs follow a pseudo-linear pattern.

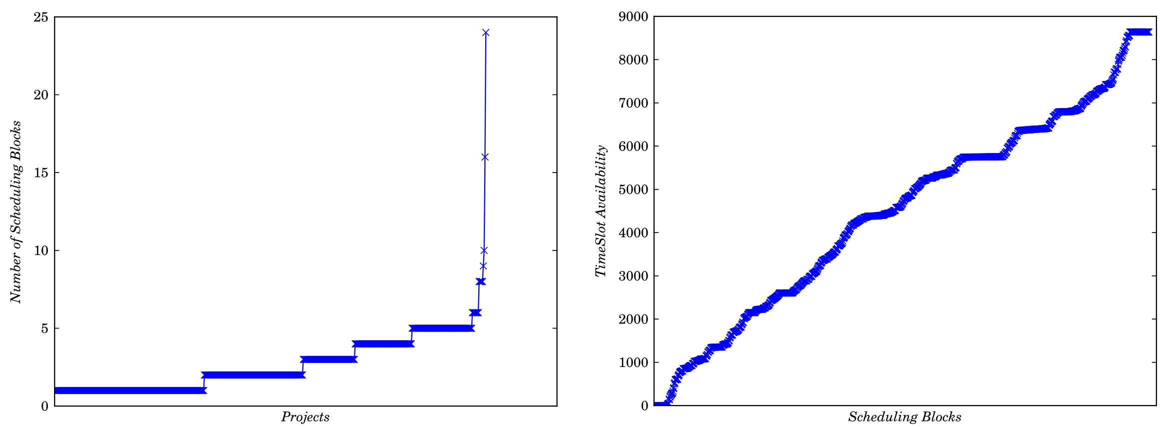
Concerning the composition of the instance files, each instance is composed by an adapted text file in which the observational season, observational projects and their specific information are literally described. In particular, initial SB time availabilities can be calculated in advance by using information like target coordinates (RA/DEC) and popular astronomical mathematical procedures and so, time slots can be generated by splitting all SB feasible times in equal-sized time lapse of 30 min and later, allocating each



(a) Distribution of SBs among projects.

(b) Distribution of time slots among SBs.

Fig. 5. Main distributions in ALMA Cycle 0 operations.



(a) Distribution of SBs among projects.

(b) Distribution of time slots among SBs.

Fig. 6. Main distributions in ALMA cycle 1 operations.

split lapse of time in their corresponding time slot representation. Also, for simulation purposes, all weather data is intended to be known in advance so, all feasible Time Slots that not meet weather constraints are filtered. Finally, SB duration is considered to have a static length of 30 min.

4.2. Execution parameters

The implementation of the proposed algorithm has a wide variety of parameters to change the behavior of the algorithm. CPLEX adds the possibility to modify internal parameters of the heuristics that are being used, but other parameters linked specifically to the algorithm can also be modified. The main parameters that are considered to be modified for the execution of the algorithm are described below:

- Long Term Time Slot Size (lts): Corresponds to the size of each long-term TimeSlot used to calculate a long-term schedule.
- Size of Gliding Window (h): Value of the size of the short-term time window used to calculate every short-term schedule.
- Hop of Gliding Window (s): Corresponds to the value of executed SBs between each short-term schedule process.
- Limit Time of Each Subproblem: Value of CPLEX time limit of each subproblem. If a subproblem could not obtain the optimal solution in the specified time then the best feasible solution is considered.
- Number of CPU's: Number of CPU's used for the parallelization of the B&C process of each subproblem. By default, all CPU's are used.

4.3. Results

This section describes the obtained results by executing the proposed scheduling algorithm using several real scale instances. Tests were performed using OSX 10.9 (Unix based) with Clang-500.2.79 compiler (C++) and CPLEX Optimization Studio v12.5 (CPLEX, 2014). About the main hardware characteristics, an Intel Core i7 4850HQ CPU clocked at 2.3 GHz and 16 GB of RAM were used to run the algorithm.

4.3.1. Instances and configuration

The results of the execution of the proposed scheduling algorithm using several generated instances divided in two main scenarios (Scen1 and Scen2) are detailed in Table 3. What both scenarios have in common is that they allow the study of the behavior of the algorithm on a projected situation of ALMA in which the demand to use the observatory is bigger than the available observation time. Inevitably, a percentage of observational projects will not be executed (in practice those projects are transferred to the next season with a higher priority).

Also, scientific priority rank scores were generated using a standard uniform distribution between 0,0 and 7,0. On the other hand, the algorithm parameter configuration used to execute all instances are detailed below:

- Long Term Time Slot Size (lts): 48 [TimeSlots] (24 h)
- Size of Gliding Window (h): 96 [TimeSlots]
- Hop of Gliding Window (s): 48 [TimeSlots per Iteration]
- Limit Time of Each Subproblem: 30 min
- Number of CPU's: All Available.

Table 3
Characteristics of real-scale instances per scenario.

	Scen1	Scen2
Total observation time (months)	3	6
Number of projects	2.500	5.000
Number of SBs	5.000	10.000
Number of SBs with precedence	0	0

Table 4
Summary of results of scenario 1 instances.

	ALMA cycle 0		ALMA cycle 1	
	Average	σ^2	Average	σ^2
ST	8591	96.84	8538.2	57.27
$\eta(TS)$	99.958%	0.0004	99.949%	0.0003
N.O.S.	0%	0	0%	0

Table 5
Summary of results of scenario 2 instances.

	ALMA cycle 0		ALMA cycle 1	
	Average	σ^2	Average	σ^2
ST	17259.4	100.22	17198	160.02
$\eta(TS)$	99.94%	0.0284	99.89%	0.049
N.O.S.	0%	0	0%	0

Table 6
Execution time for scenario 1 instances.

	Cycle 0		Cycle 1	
	Average	σ^2	Average	σ^2
t_{total} (s)	2032.66	40.20	1622.03	46.98
$t_{Short-Term}$ (s)	22.58	0.44	18.02	0.52

Table 7
Execution time for scenario 2 instances.

	Cycle 0		Cycle 1	
	Average	σ^2	Average	σ^2
t_{total} (s)	13014.36	241.72	10524.60	190.79
$t_{Short-Term}$ (s)	72.30	1.34	58.47	1.05

4.3.2. Obtained results

Using both Cycle 0 and Cycle 1 characteristics, five different instances were used for each scenario. After executing the algorithm over all generated instances, a summary of the obtained results by measuring the scientific throughput (ST) (Eq. (5)) and the number of Non-Optimal Solutions (N.O.S.) obtained in each short-term schedule process are detailed in Table 4 for Scenario 1 and in Table 5 for Scenario 2. Also, the Time Slot Allocation Efficiency ($\eta(TS)$) (Eq. (13)) is detailed on each table to analyze the precision of allocation of the algorithm.

$$\eta(TS) = \left(1 - \frac{WastedTS}{TotalTS - NotUsedTS}\right) \times 100\% \qquad (13)$$

where $WastedTS$ is the sum of time slots assigned to SBs of uncompleted projects and $NotUsedTS$ the number of time slots not used at the end of the schedule process.

Execution times of the algorithm were also measured by not only considering the total execution time of each instance but also, the average execution time of each Short-Term schedule. A summary of the execution time of the algorithm over the described instances are detailed in Tables 6 and 7 for scenario 1 and 2, respectively.

Finally, for all test executions, the highest CPU load average obtained was around 1.97/8 and the average RAM usage was around 1.28 GB.

Table 8
Main characteristics of instances to evaluate the performance.

Total observation time	1 month
Number of projects	500
Number of SBs	1000
SBs with precedence	0 and 640

4.4. Algorithm performance

Some tests to evaluate the performance of the proposed algorithm were made. Considering that the astronomical scheduling problem is NP-Hard, using real scale instances can take a lot of time and computer resources to solve, specially when the optimal solution of the problem is desired to be found. According to the previous statement, more simple instances were generated in order to get the optimal solution (using an optimal solver) in a reasonable time while not getting out of computer resources during the solving process. In that way, in Table 8 instances used to evaluate the performance of the algorithm are described.

4.4.1. Objective function

The performance of the proposed algorithm by studying behavior of the scientific throughput (Eq. (5)) is analyzed in this section. To perform the analysis, an optimal solver was implemented to measure the gap between the obtained objective function value and the optimal solution. The optimal solver was constructed using CPLEX (2014) with the same constraints described in Section 3.3 and defined explicitly in the model the objective function detailed in Eq. (5).

Considering the optimal values obtained by running all test-scale instances, the average relative gap (defined in Eq. (14)) between the optimal values and the obtained by different configurations of h and s parameter are detailed in Fig. 7.

$$RelativeGap(x) = \left(\frac{x - OptimalValue}{x}\right) \cdot 100[\%]. \qquad (14)$$

In Fig. 7(a), different values of h were tested by setting statically $s = 1$ and $lts = 48$, and in Fig. 7(b) different values of s were tested by setting $h = 96$ and $lts = 48$.

4.4.2. Execution times

In this section, the execution time of the algorithm is analyzed. In the same way as the objective function was analyzed, the total elapsed time that the algorithm takes to run each instance was analyzed and detailed in Fig. 8. In Fig. 8(a), the relation of the total execution time of the algorithm in terms of h is shown by setting $s = 1$ and $lts = 48$ and in Fig. 8(b), different values of s were tested by setting $h = 96$ and $lts = 48$.

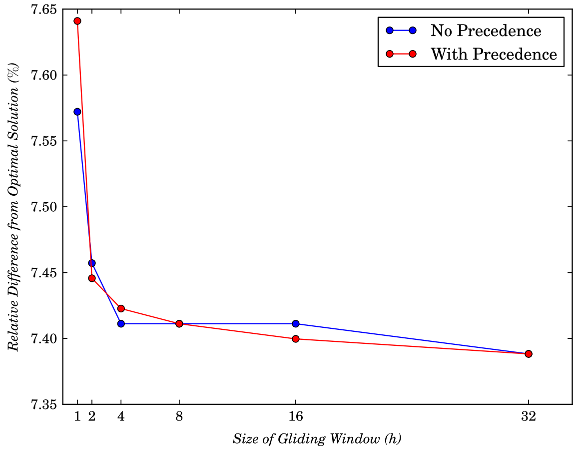
Following the same methodology, in Fig. 9 the relation of the average elapsed time of each short-term schedule process (or algorithm iteration) with h and s parameters are shown.

4.4.3. CPLEX model attributes

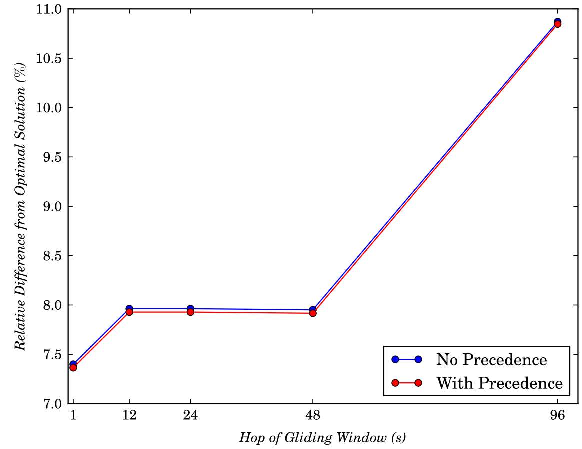
Some specific attributes of the short-term schedule model are also analyzed. In first place, the total number of CPLEX variables created to solve each short-term schedule was analyzed using the same methodology as other analysis in previous sections and the results are detailed in Fig. 10.

In the same way, the total number of constraints created by CPLEX to solve each short-term schedule are detailed in Fig. 11.

Finally, the total number of nodes used by CPLEX was analyzed. In Fig. 12 the total number of nodes are detailed by varying s and h parameter.

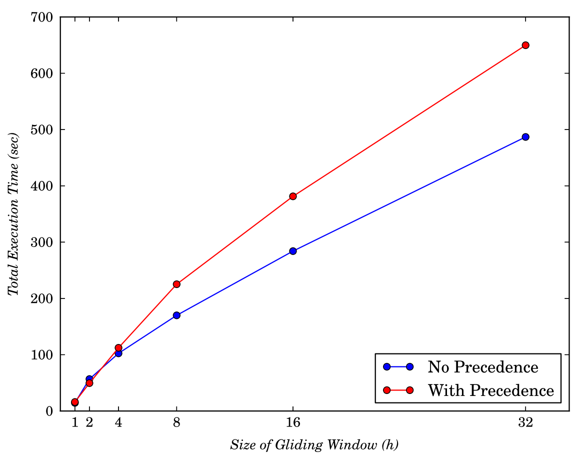


(a) Varying h parameter.

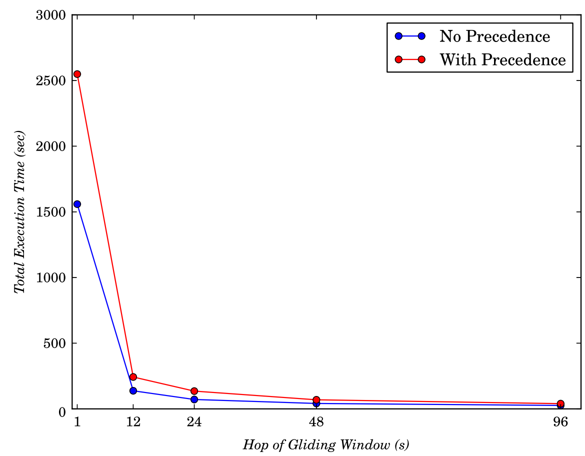


(b) Varying s parameter.

Fig. 7. Average gap from optimal *ST* (less is better).

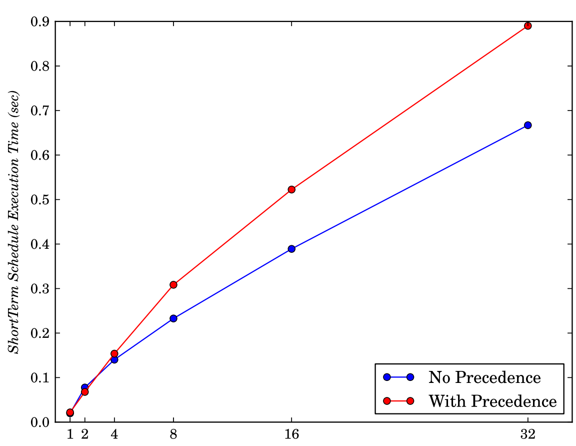


(a) Varying h parameter.

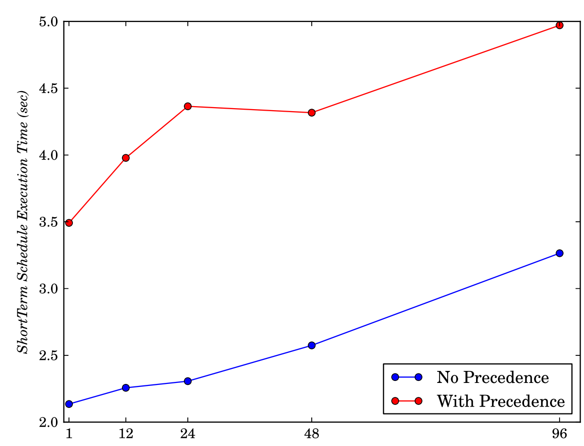


(b) Varying s parameter.

Fig. 8. Total execution time (less is better).



(a) Varying h parameter.



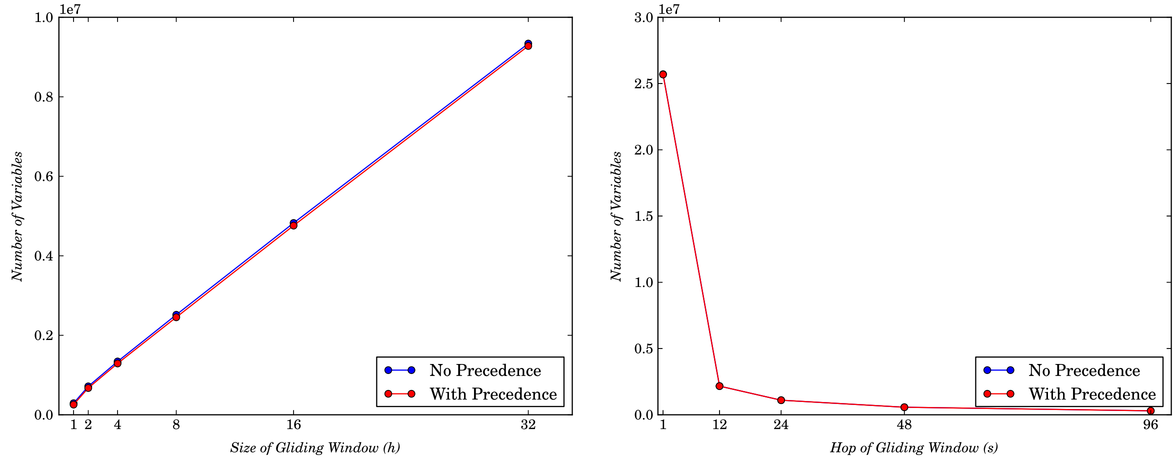
(b) Varying s parameter.

Fig. 9. Average short-term schedule execution time (less is better).

4.5. Analysis of results

According to the results obtained by executing the algorithm over real scale instances (see Section 4.3.2), some important observations can be highlighted:

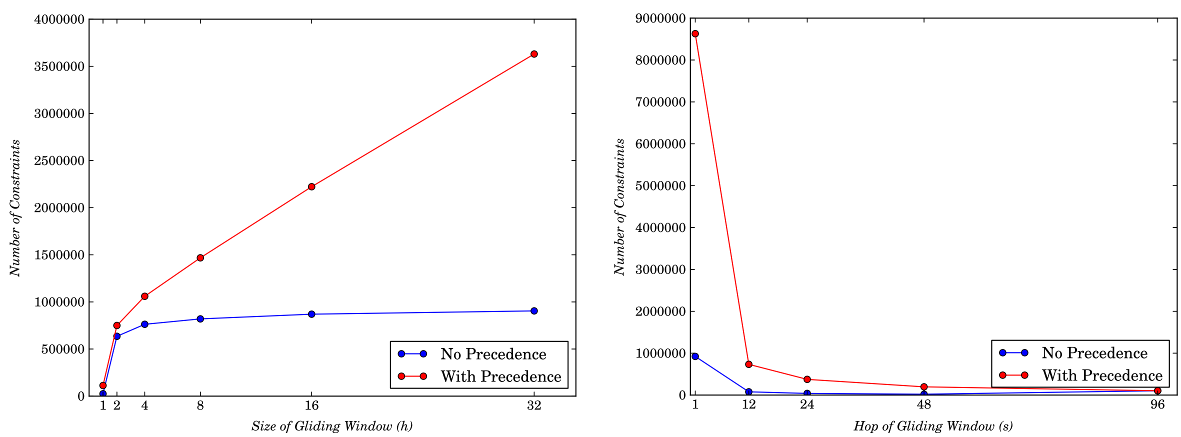
- The algorithm solved real scale instances with a high rate of time slot allocation efficiency (Eq. (13)) which means only a few percentage of observation time is wasted because of bad scheduling decisions in the scheduling process of the algorithm.
- In a real application scenario the scheduling algorithm will run in an online mode (as Fig. 2 shows) and so, one of the more



(a) Varying h parameter.

(b) Varying s parameter.

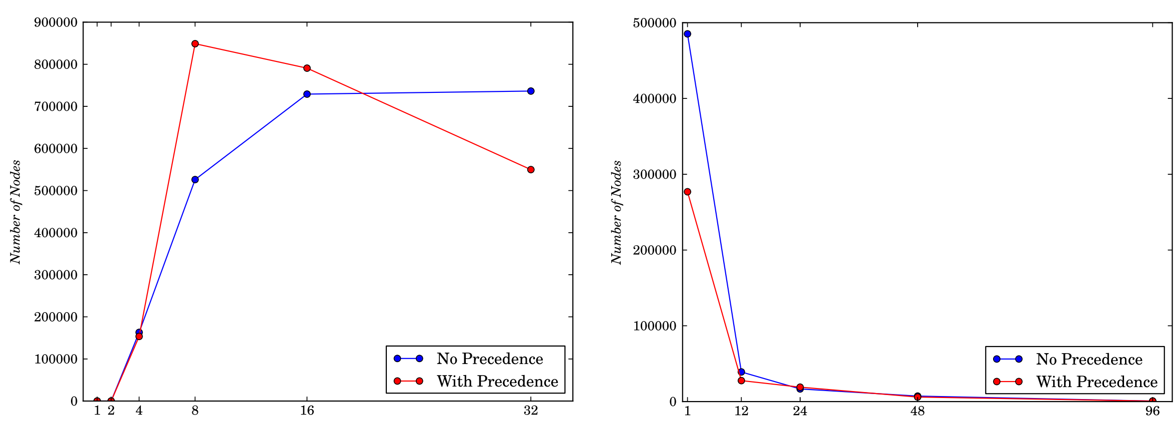
Fig. 10. Number of total variables created.



(a) Varying h parameter.

(b) Varying s parameter.

Fig. 11. Number of total constraints created.



(a) Varying h parameter.

(b) Varying s parameter.

Fig. 12. Number of nodes explored.

important variables is the reaction time of the next schedule plan for the current short-term period. In that way, the obtained results show that the reaction time of the algorithm for each short-term schedule (or re-schedule if any dynamic condition changes) is around 1 min in the heaviest tests (Tables 6 and 7). On the other side, a complete schedule simulation took about an hour in the heaviest test to be completed.

- In all execution tests, non-optimal short-term schedule solutions have not been found. This means that in general scheduling subproblem is not too complex and so, a heavier short-term schedule configuration can be made in order to optimize even more the scientific throughput (Eq. (5)).

According to the results obtained by running the algorithm over test-scale instances (Section 4.4), the following statements can be made:

- Average gap between the optimal objective function and the obtained through running the algorithm with different configurations (Fig. 7) shows that the best average gap from the optimal solution was around 7.4%. Also, Fig. 7(b) reveals that whenever the *s* parameter increases (letting constant *h* parameter) the objective function decreases.
- There is a direct relation between the size of the gliding window used to solve each short-term schedule problem and the objective function value. On the other hand, there is an inverse relation between the hop of the gliding window and the objective function value.
- There is an inverse relation between the size of the gliding window used to solve each short-term schedule problem and execution times. In the same way, there is a direct relation between the hop of the gliding window and the execution times. Both relations follow a non-linear pattern as might be expected for an NP-Hard problem type.
- In most cases there is a difference in the results between instances with and without precedence constraints. This difference reveals that the performed instances with precedence constraints are more difficult to solve (in average) due the extra amount of time needed to execute them, although, this difference is small.

5. Conclusions

This study has proved that the scheduling problem in astronomy is a very challenging problem and there is still enough space to contribute. Even when the proposed algorithm takes advantage of some assumptions that simplifies the real problem, this work was able to produce a mathematical model and a prototype implementation of a scheduling solution that solves successfully a simplified version of ALMA's scheduling problem considering projected volumes of demand for the next years.

This scheduling algorithm features a multi-layer model that performs a schedule process in both long-term and short-term phases. Also, a time discretization and both static and dynamic variables are considered in this scheduling model in which Scheduling Blocks are arranged over the planning time thanks by a MILP-based solver that performs the core task of the scheduling process.

The main objectives detailed in Section 1 were accomplished successfully, while their respective results can be checked as follow:

1. A list of static and dynamic parameters, and scheduling constraints of ALMA observatory was researched to build the proposed scheduling algorithm.
2. A mathematical scheduling model based on a MILP approach to solve the simplified ALMA's scheduling problem was proposed taking in consideration the identified variables, constraints and assumptions described in Section 2.
3. A prototype of the proposed algorithm was programmed in C++ using CPLEX framework.
4. ALMA scheduling data of both Cycle 0 and Cycle 1 was characterized and used to simulate data and test the proposed algorithm.
5. The results of the gap of the obtained solution from the optimal, execution times and internal model attributes was presented by executing synthetical instances over the implementation prototype.

According to the obtained results we conclude the following:

- Tested instances reveal that the algorithm achieves an acceptable performance according to the obtained percentages of time slot allocation efficiency (Eq. (13)) and the relative gap between the calculated objective function value and the optimal one (Eq. (14)). Also, results reveal that this gap can be minimized even more by tuning the algorithm with a higher value of *h* parameter.
- Under a real case scenario with the proposed execution parameters, the implementation of the algorithm takes a considerable amount of time and resources compared to other scheduling solutions in the literature. Despite that, the established parameter configuration was able to solve all real scale instances in a feasible amount of time without using any sophisticated hardware environment. On the other hand, this behavior may be reversed if another parameter configuration is used (with a possible yield decreasement).
- An aspect that is extremely important in practice is the reaction time of the algorithm if a re-schedule is needed due to any changes in the current conditions that affect the scheduling process. In that way, the results showed that the proposed algorithm needs about a minute for a re-schedule in the worst case tested (considering the used parameters and instance scales).
- Instances with precedence constraints took more time to be solved than those without precedence constraints. At some point it should not be strange that this relation can be inverted due the well known statement in complexity theory about the relation between the number of constraints and the complexity of a problem. Unfortunately, this last behavior was not evidenced in the performed tests.

Finally, this work provides an algorithm solution that can be improved and also motivates to continue researching in more algorithms and heuristics applied to the scheduling in astronomy in which there is a huge field that is still uncovered.

Acknowledgments

This article was partially funded by the project ECOS-Conicyt #C10E02, and DGIP-UTFSM.

References

Actis, M., et al., 2012. Design concepts for the Cherenkov telescope array CTA: an advanced facility for ground-based high-energy gamma-ray astronomy. Exp. Astron. 32 (3), 193–316.
Avarias, J., 2014. A new scheduling algorithm for ALMA observatory. In: Master Thesis in Computer Science. Universidad Técnica Federico Santa María, Valparaiso, Chile.
Balser, D.S., et al., 2009. GBT dynamic scheduling system: Algorithms, metrics, and simulations. In: Astronomical Society of the Pacific Conference Series, vol. 411. p. 147.
Bridger, A., Butler, B., 2008. The ALMA/EVLA project data model: steps toward a common project description for astronomy. Proc. SPIE 7019, 70190P–70190P–11.
Bucher, S., et al., 2008. Chandra mission scheduling on-orbit experience. 70160W–70160W–12.
Buchner, J., 2011. Dynamic Scheduling and Planning Parallel Observations on Large Radio Telescope Arrays with the Square Kilometre Array in Mind. In: Master of Computer and Information Sciences, School of Computing and Mathematical Sciences. Auckland University of Technology, New Zealand.
Castro, P.M., Grossmann, I., 2006. An efficient {MILP} model for the short-term scheduling of single stage batch plants. Comput. Chem. Eng. 30 (6–7), 1003–1018.
Chang, G.W., et al., 2001. Experiences with mixed integer linear programming based approaches on short-term hydro scheduling. IEEE Trans. Power Syst. 16 (4), 743–749.
Civeit, T., Automated long-term scheduling for the SOFIA airborne observatory, in: Aerospace Conference IEEE, 2013, pp. 1–7.
Clarke, D., Avarias, J., 2012. ALMA scheduling: It's dynamic!. In: Astronomical Data Analysis Software and Systems XXI. In: ASP Conference Series, vol. 461. Astronomical Society of the Pacific.
Colome, J., et al. 2010. The TJO – oadm robotic observatory: the scheduler.
Colome, J., et al. 2012. Research on schedulers for astronomical observatories. 8448:84481L–84481L–18.
IBM ILOG CPLEX Optimization Studio. Software (2014). http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/.

De Jong, A., 2012. Scheduling on the low frequency array (lofar) telescope. In: Astronomical Data Analysis Software and Systems XXI, Vol. 461.
Floudas, C.A., Lin, X., 2005. Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. Ann. Oper. Res. 139 (1), 131–162.
Garcia-Piquer, A., et al., 2014. CARMENES instrument control system and operational scheduler. In: Proc. SPIE Software and Cyberinfrastructure for Astronomy III, Vol. 9152. p. 915221. doi:10.1117/12.2057134.
National Radio Astronomy Observatory (NRAO) (2013). Observing With The Green Bank Telescope. Observer Guide Manual.
Gómez de Castro, Ana I., Yáñez, J., 2003. Optimization of telescope scheduling. Astron. Astrophys. 403 (1), 357–367.
Graham, R.L., 1966. Bounds for certain multiprocessing anomalies, vol. 45.
Granzer, T., 2004. What makes an automated telescope robotic? Astron. Nachr. 325 (6–8), 513–518.
Grim, R., et al., 2002. Fp. Proc. SPIE 4757, 51–61.
Hiriart, R., 2010. ALMA Dynamic Scheduling Algorithm. Tech. Rep. National Radio Astronomy Observatory, Socorro, NM.
Hoffstadt, A., 2010. Planning Mode Simulator: A Simulation Tool for Studying ALMA's Scheduling Behavior (undergraduate thesis). Universidad Técnica Federico Santa María, Valparaiso, Chile.
Ierapetritou, M.G., Floudas, C.A., 1998. Effective continuous-time formulation for short-term scheduling. 1. multipurpose batch processes. Ind. Eng. Chem. Res. 37 (11), 4341–4359.
Johnston, M.D., 1988. Automated Observation Scheduling for the VLT [microform] / Mark D. Johnston. Space Telescope Science Institute Baltimore, MD.
Johnston, M.D., Adorf, H., 1992. Scheduling with neural networks—the case of the Hubble space telescope. Comput. Oper. Res. 19 (3–4), 209–240.
Johnston, M.D., Miller, G.E., 1994. Spike: Intelligent scheduling of Hubble space telescope observations. In: Intelligent Scheduling. Morgan Kaufmann Publishers, pp. 391–422.
Lee, H., et al., 1996. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. Ind. Eng. Chem. Res. 35 (5), 1630–1641.
McCarty, M., et al., 2012. The GBT dynamic scheduling system. In: Astronomical Society of the Pacific Conference Series, vol. 411. p. 147.
Mora, M., 2011. A Scheduling Algorithm with Dynamic Priorities (Master's thesis). Universidad Técnica Federico Santa María, Valparaiso, Chile.
Mora, M., Solar, M., 2010. A survey on the dynamic scheduling problem in astronomical observations. In: Bramer, M. (Ed.), Artificial Intelligence in Theory and Practice III. In: IFIP Advances in Information and Communication Technology, vol. 331. Springer, Berlin Heidelberg, pp. 111–120.
Nyman, L.A., et al., 2010. ALMA science operations. Proc. SPIE 7737, 77370G.
Puxley, P., 1997. Execution of queue-scheduled observations with the gemini 8m telescopes. In: Lewis, H. (Ed.), Telescope Control Systems II. In: Proc. SPIE, vol. 3112. pp. 234–245.
Sebag, J., et al., 2007. LSST Cloud Cover Measurement for Observation Scheduling. In: Revista Mexicana de Astronomia y Astrofisica Conference Series, vol. 30. p. 35.
Sessoms, E., et al., 2009. The GBT dynamic scheduling system: Scheduling applications of the knapsack problem and sudoku. In: Astronomical Society of the Pacific Conference Series, vol. 411. p. 351.
Silva, D.R., 2002. Service mode scheduling at the ESO very large telescope. In: Quinn, P.J. (Ed.), Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 4844. pp. 94–103.
Spotts, P., 2010. New telescopes could revolutionize astronomy, but at what price?.
Watson, M., Cacioppi, P., 2014. A Deep Dive into Strategic Network Design Programming: OPL CPLEX Edition. Watsoppi Press, p. 31.
Wetter, J., et al., 2015. Cloud-based e-infrastructure for scheduling astronomical observations. In: IEEE 11th International Conference on eScience, vol. 362. p. 370.