INTERNAL REPORT

Towards a dynamic scheduling system for the SRT

F. Buffa, G. Serra, G.L. Deiana

Report N. 67, released: 20/06/2017

Reviewer: A. Tarchi

OAC Osservatorio Astronomico di Cagliari

Acronyms

ALMA	Atacama Large Millimeter Array
ANN	Artificial Neural Network
AOP	Antenna Operative Model
CSP	Constraint Satisfaction Problem
DB	Database
DS	Dynamic Scheduling
ESO	European Southern Observatory
GBT	Green Bank Telescope
GA	Genetic Algorithms
HST	Hubble Space Telescope
JSSP	Job-Shop Scheduling Problem
KP	Knapsack Problem
LTS	Long Term Schedule
NP	Non-deterministic Polynomial time
PSS	Proposal Submission System
RFI	Radio Frequency Interference
SA	Simulated Annealing
SKA	Square Kilometer Array
SRT	Sardinia Radio Telescope
STS	Short Term Schedule
TAC	Time Alloca1tion Committee
ToO	Target of Opportunity
Tsys	System Temperature
VLA	Very Large Array
VLBI	Very Long Base Interferometry
VLT	Very Large Telescope
WRF	Weather Forecast Model

Towards a dynamic scheduling system for the SRT

"NP-hardness = intractability"
Silvano Martello and Paolo Toth

"The future is independent of the past, given the present"
Andrej Markov

Abstract

The implementation of the dynamic scheduling for the allocation of the scientific observations is a key step to efficiently exploit the capabilities of the Sardinia Radio Telescope to observe, in particular, at the higher frequencies. In fact, thanks to the SRT agility to switch remotely its different receivers, an opportune allocation of the high frequency observations, taking into account the weather conditions and the forecast of the atmosphere status over the facility site, would certainly increase the scientific productivity.
In this technical report, we propose a dynamic scheduling algorithm which, by gathering the outcomes of different sub-systems already available at the SRT site (weather forecast model and atmospheric sensors), is able to processes, 48-hours in advance, a schedule of tasks taken from a pool of different activities (scientific observations, planned maintenance and so on) and maximize the exploitation of the antenna.
This dynamic scheduling algorithm (scheduler), based on the knapsack problem formulation, has been tested by simulating a 48-hours long schedule. The results of these tests are shown and discussed.

1 Introduction

Dynamic scheduling (hereafter DS) is the allocation to competing jobs of shared resources in order to maximize their exploitation. This is an optimization problem which produces a significant amount of scientific literature aimed at proposing efficient algorithms to solve it in different application areas [1].
In computer science [2], DS algorithms are applied to determine in real-time which instruction has to be executed in concurrent programming in order to optimize the use of the available resources (operands, memory and CPU cores).
In astronomy, the implementation of the DS algorithms offers the opportunity to take as much as possible advantages from a long-term public investment, as the high sensitive telescopes, by efficiently allocating their observations [3, 4]. In this case, DS algorithms can help to issue a schedule which maximizes the telescope observing time by trying to fulfill the scientific goals, which such a resource has been built for,

and taking into account all the competing tasks/jobs needed for the telescope operation. The planned maintenance, relevant and unmissable astronomical observation of a cosmic target (e.g. Target of Opportunity, ToO) and the international projects with other telescopes (e.g. Very Long Base Interferometry, VLBI) are only some examples of competing jobs which can share the resource-telescope with the worthy observations selected by means of an international call for scientific proposals.

In addition, the telescope exploitation can be improved if the DS algorithms take into account even the limiting environment factors, as the atmosphere variability, which might affect the run of competing jobs. In order to reduce the number of the scheduled jobs not successfully completed due to these environment factors, DS algorithms can be supported by the outcomes of a weather forecast model and the site monitoring system. This union defines a DS system whose implementation is desirable at the radio astronomical sites, where telescopes operate in a wide frequency range by using many receivers. As a matter of the fact, the operation of the array of radio telescopes [5, 6] and large single-dish antennas [7] benefits from a DS system because observation frequency (i.e. the receiver) can be decided in advance thanks to the atmosphere status forecast at the telescope site. In this way a decisional algorithm might first minimize the telescope down-time and then choose the observation which better fits to the expected weather and atmosphere conditions.

However, although the DS algorithms can achieve a high degree of automation, the human supervision is always needed for last minute decisions, e.g. when an external cause perturbs the current schedule, and it has to be considered in the algorithm design phase.

We can conceive the SRT DS general scheme such as the one depicted in Figure 1. The system is integrated into the Antenna Operative Model (AOP) with the aim to optimize the allocation of the observations, exploiting the weather forecast model (WFM) and the site sensors (radiometer, weather station, monitoring system for the radio frequency interference, RFI).

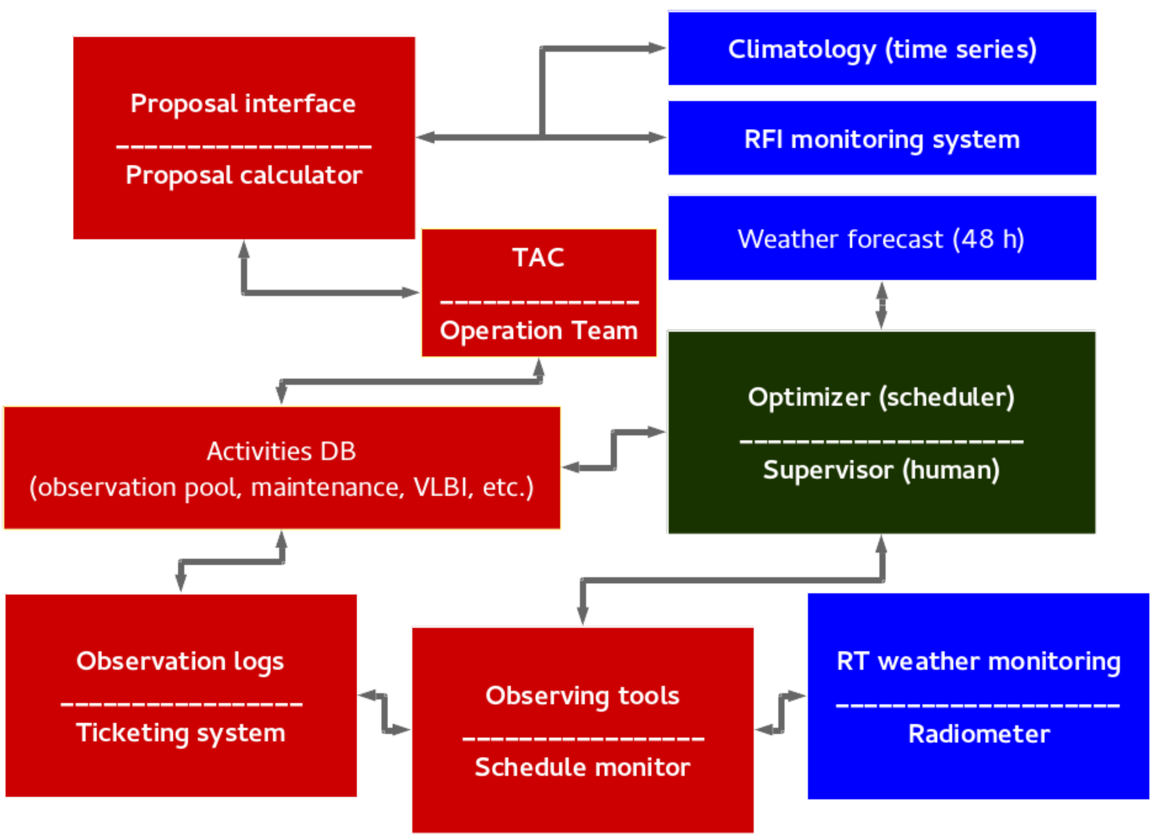

Fig. 1. Scheme of a possible dynamic scheduling system integrated into the antenna operative model, the optimizer (dark green box) is the main topic of this report.

The block diagram shows the DS components from the calculator for supporting the proposals submissions (interface and calculator) to the antenna activity database (DB). The optimizer/supervisor considers all these resources (material and human) in order to release a new schedule. The schedule optimization is based on a scoring criterion which is driven by technical (weather, RFI) and scientific (Time Allocation Committee indications) factors.
Once the schedule is issued, the system provides the observer with tools for monitoring the schedule and for reporting the tickets and the logs of each observation.
Nevertheless the main topic of this report is the schedule optimization (the Optimizer of Figure 1).
After a short *excursus* among the main scheduling approaches applied at other radio-telescopes in Section 2, a scheduler implementation based on the knapsack problem formulation is presented in Section 3. Moreover, in Section 3 a simulated annealing approach is proposed to solve this kind of problem.
Then, in order to assess the proposed scheduler, a 48-hours long schedule is simulated and results are shown in Section 4. Finally, in Section 5, discussion on the DS and future developments are outlined.

2 Scheduling approaches at other radio-telescopes

The scheduling approaches at the radio-astronomical facilities around the world can be mainly distinguished into two categories depending on the available resources: single-telescope/multi-receivers (single-dish category) and multi-telescopes/multi-receivers (array category). The SRT and the Green Bank Telescope (GBT) belong to the single-dish category. The Atacama Large Millimeter Array (ALMA) and the Very Large Array (VLA) belong to both categories, since they are typically used as an array, but also, although more rarely, as a group of antennas. In this latter case the optimization approach must take into account a certain degree of parallelization.
These two main approaches exploit generally two different algorithms based on the *knapsack* and the *job-shop* optimization problem, respectively.

2.1 Knapsack problem

In the single-dish category the scientific observations are in a queue of jobs which the telescope has to run in an order such that the telescope exploitation is maximum taking into account both the competing jobs and the limiting factors. This kind of problem is called conventionally knapsack problem (KP), see Figure 2. KP is a well-known and widely studied computational problem in the combinatorial optimization field. Its general formulation can be expressed as:

"A thief, having a bag with a capacity N, has to evaluate a number M of valuables each one having a different weight (w) and value (v). The thief can only carry objects for a maximum weight W".

In other words, the question for the algorithm implementation is: what is the optimal assortment of objects such that the thief maximizes the value of objects in his bag with the weight constraint W? The answer can be found by maximizing the following objective function:

$$f = \sum_{j=1}^{M} x_j v_j \tag{1}$$

where f is the loot, v_j is the object value and x_j is 1 if the object has taken and put in the bag, 0 otherwise; with the following constraints:

$$\sum_{j=1}^{M} x_j w_j \leq W \text{ , } x_j \in \{0, 1\} \text{ and } N \leq M$$

depending on the weight w_j of each valuables, the weight W and the capacity N of the bag.

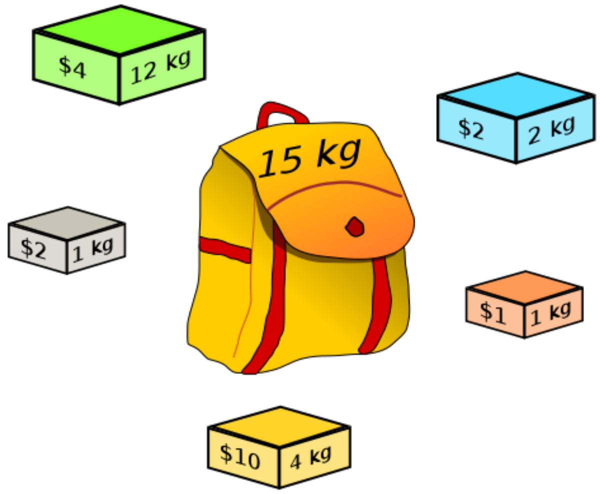

Fig. 2. Knapsack problem example: the loot is maximum with three yellow boxes and three grey boxes, if any number of each box is available, otherwise with all the boxes except the green box, if only shown boxes are available (from Wikipedia).

Ceteris paribus, by applying the KP to the single-dish scheduling, the observations (the valuables), having a given time span (the weights), have to be evaluated in terms of their scientific grade (the value) and in terms of telescope time allocation (the capacity of the bag). Therefore, the goal of the scheduler (the thief) is to determine the observation sequence which maximizes the telescope scientific score (the robbery loot) taking into account that the observation score can vary with the atmosphere status or every contingencies (technical failure).

2.2 Job-shop problem

In the array category the observations may be concurrent and the optimization algorithm might select many observation sequences and harmonize them to exploit all the available telescopes. The job-shop scheduling problem (JSSP) consists basically on:

"*A set of M jobs has to be run in parallel on a set of N machines by choosing the sequence of the job run which minimize the time required to complete the sequence (makespan)"*.

Therefore, the scheduler's goal is to find out the schedules which minimize the makespan.
Since the Job-shop problem implementation goes beyond the goal of this report, the authors invite the interested reader to refer to technical bibliography in order to get more details about it [8].

2.3 Examples of DS application

Ground-based, space or satellite facilities developed their own approaches to implement the more suitable DS for their own purposes with a different level of automation: from a human-decision-based approach to fully-automated one. In this Section some practical examples of algorithm implementation are discussed. The selected examples are not necessarily the most significant or the most efficient from a computational point of view, they are instead representative of the many possible approaches that one can consider in the algorithm design phase.

2.3.1 Hubble Space Telescope

The Hubble Space Telescope (HST) scheduling system (SPIKE) is based on a constraint satisfaction problem (CSP) algorithm [4]. The computer code deals with a complex scheduling problem involving the allocation of a number of observing slots ranging from 10000 to 30000 per year.
In this example, a long- and short-term scheduling concept is generally applied: the long-term scheduler allocates a set of activities to a period which can last up to a week; the short-term one is in charge of the day-by-day activities allocation within this period. The scheduler core is a heuristic algorithm implementing the following main actions:

i) find a best-first solution (trial assignment);
ii) heuristic search and solution of possible constraint violations (repair);
iii) iterative resolution of conflicting activities (avoid conflict).

The SPIKE optimization steps and the scheduling choices are supervised by an Artificial Neural Network (ANN).

2.3.2 GERRY

GERRY was a DS system implemented for the scheduling of the Space Shuttle ground operations [4]. Three types of constraints are considered:

i) priority between activities;
ii) usage of resources;
iii) activity requirements.

A weighted penalty function is used to measure the cost of constraint violation. The repair procedure considers each type of constraint independently. In order to avoid trapping at local optima in the search space, they use simulated annealing to determine the acceptance of a newly generated schedule. To solve resource constraints, the tasks are selected for repair (schedule optimization) considering three heuristic criteria:

i) move the task on the basis of resource requirements (fitness);
ii) move the task having the fewest temporal dependencies (dependency);
iii) move the task which needs the smallest move to resolve the conflict (logistic).

The task with the highest score with respect to these criteria is then selected. The GERRY scheduler turned to be very effective and it was incorporated into the NASA Ground Processing Scheduling System, a tool for scheduling repair and refurbishment of the space shuttle missions.

2.3.3 Square Kilometer Array

The SKA requirements for a DS scheduling are extensively analyzed in a recent PhD thesis [6]. This work considers dynamic and scalable re-scheduling, management of many resources, parallel observations and dynamic parallel observations. Due to the complexity of the problem, exact algorithms such as branch-and-bound turned out to be time-consuming. The fastest of these algorithms, Linear (Integer) Programming, required an unreasonable amount of time (days) to provide unsatisfactory solutions. At the end, a DS scheme, based on Genetic Algorithms (GA), able to allocate the highest-priority observation possible at each time-slot is proposed as a candidate for SKA.

2.3.4 ESO Very Large Telescope

At the beginning the HST SPIKE [3] was used at ESO for the Very Large Telescope (VLT) and adapted to its scheduling requirements. In this case a human-based scheduling was supported by an automatic scheduler to mix both visitor and service mode observations[1].
Recently, a new scheduler based on the concept of artificial intelligence were introduced with the aim to get significantly reductions in calculation time. The new code takes into account many factors affecting the observations, such as weather and technical down-times. Observations are generally gathered and then scheduled during a period of 6 months.

2.3.5 ALMA

ALMA operates exclusively in service mode and the Scheduling Subsystem provides a fully automated quasi-real-time dynamic allocation of the observations [5].
The priority of the dynamic of the scheduling is the main problem, which slightly differs from the traditional job-shop approach. The priorities might vary, depending on the internal and the external conditions and, as a consequence, the observing schedule has to be changed to fit the current system and the environmental status. Once a list of candidate blocks which satisfy the source visibility and the execution time limit criteria is created, the probability that each scheduled block reach its scientific goal is calculated. The ranking of the candidates depends on the scientific priority, setup time, resources consumed, number of scheduling blocks not still completed, and so on. For each observation block a final score is computed by multiplying the block ranking and its probability being successfully completed.

2.3.6 GBT

The GBT DS system schedules the jobs from 24 to 48 hours in advance, taking into account the highly variable weather patterns of the telescope site [7]. The observer availability and his/her qualification for a remote (service) observation, weather forecasts[2] and the telescope status, are the main constraints for the DS implementation. If there are periods of time or dates when the observer cannot perform observations, he can indicate the “blackout dates” in the DS interface.

---

[1] Traditionally astronomers travel to the telescope facility to observe (visitor mode). In service mode the telescope is operated by the local staff and the astronomers perform/supervise its observation remotely.
[2] GBT adopted the forecasts provided by the National Weather Service with 12 km spatial resolution, http://www.wdtb.noaa.gov/tools/BUFKIT/.

The system includes a resource calendar to consider the telescope maintenance activities and a Sensitivity Calculator supporting the observers for the proposal preparation.
The GBT DS implementation is based on a variant of the knapsack algorithm applied only to the subset of the competing observations. Conversely, many of the GBT observations are either fixed (VLBI sessions) or windowed. Fixed observations must be executed at a specific epoch for a specific duration, while a windowed observation must be run within a limited temporal span.
Currently, 40% of GBT time is consumed by windowed activities [9]. Fixed and windowed observations need to be scheduled with constraints and without optimization. The GBT DS algorithm looks like a Sudoku puzzle, where the board corresponds to the schedule and the prefilled blocks correspond to the fixed sessions. Its implementation is based on the recursive Floyd-Warshall approach which tries to fit the best achievable schedule to a shortest paths in a graph (salesman problem).

3 A possible DS algorithm for the SRT

A reliable DS system for the radio-telescope observations is possible, but it requires resources and tools to be implemented, as the GBT experience has shown.
This becomes even more perceptible for the high frequency observations (> 15 GHz) which are strongly affected by the atmosphere opacity. Over 22 GHz (frequency resonance of the water molecule) such effects become dominant and, a weather forecast model able to predict the opacity is mandatory to select the observation frequency and a decisional algorithm is desirable to dynamically schedule them.
Before the SRT shut-down for the extraordinary maintenance of its active surface, the continue weather monitoring at the site and the WFM have been successfully tested by using the SRT K-band receiver dataset [10, 11]. Moreover, the forecast capability of the WFM have been tested during the SRT early science, but still with a static scheduling of the observations.
In this Section a possible decisional algorithm which exploits the outcomes of the site atmosphere monitoring and the WFM is proposed for the SRT operation. This algorithm uses the knapsack problem formulation (see Section 2.1) and proposes a solution based on the SA approach.

3.1 Solving the knapsack problem

For the sake of simplicity, we can define two general categories of scheduling to be considered in the knapsack problem formulation:

- ❖ Long term schedule (LTS): environmental scenarios are known in statistical sense (site monitoring time series)
- ❖ Short term schedule (STS): environmental conditions are (reliably) predictable two days in advance (WFM)

LTS collects the proposed observations to be processed in the next (few) weeks. Every two days the LTS pool is filtered of those observations which cannot be performed due to whatever reason related to the antenna system. Fixed jobs/tasks (planned maintenance, VLBI runs and ToOs) are of course added to the new sub-set, but they are considered as constraints. Then, the extended sub-set contains the candidates for the final STS.
The algorithm implementation is conceptually simple: try all possible pairings of observations and environment conditions and select the ones with the highest total score. For instance, considering a 48-hour long schedule and a 30-candidates sub-set and supposing each job run takes 4 hours on average, there would be 4.14E+16 permutations to be considered. If one tried to solve the KP by means of brute-

force approach, finding a solution would be computationally infeasible. In fact, a computer capable to process one billion permutations per second would take more than 12 years. In the computer science slang the KP is defined as a non-deterministic polynomial time (NP) hard problem [12].
Both exact (deterministic) and heuristic algorithms have been proposed for the KP. Exact algorithms are essentially based on dynamic programming or branch-and-bound methods [12]. These latter allow one to find optimal solutions by means of a systematic enumeration of all the possible solutions, thus accepting a heavy computational load and a time-consuming elaboration. On the contrary, heuristic algorithms are considered able to find a (reasonably good) solution in a short time. Basically, these algorithms construct a solution by adding a new object each time into the current solution searching the local optimal choice at each iteration step.
Simulated annealing (SA) is a well-known heuristic local search algorithm which can take into account specific constraints and exceptions related to *real world* applications. Moreover, since SA is an algorithm easily implementable, capable of escaping from local optima and flexible, has been here used to solve a KP problem applied to the DS scheduling of single-dish telescopes.

3.2 Simulated annealing

Simulated annealing owes its name to the process of annealing, in which a (melted) crystalline solid is allowed to cool slowly until it reaches the minimum energy state. By mimicking this type of thermodynamic behavior, SA can be applied to the search of global optima solution in the KP.
Once an objective function has been defined, the optimal solution can be searched by varying the function parameters along a path similar to a Markov chain. At each step a new solution is generated after randomly varying one or more parameters and, the new objective function is compared to the previous one. If the new solution is better than the old one, the new solution is accepted. But not all the "not-improving" solutions are necessarily discarded, in fact they are accepted with a probability. Such a probability depends on the temperature parameter which decreases monotonically during the annealing.
Figure 3 shows a typical pseudo-code implementing the SA algorithm to solve the KP. Once defined the objective function *f*, see Equation 1, some parameters have to be set in order to drive the solution search: *T* is the pseudo-system-temperature, *a* is the annealing velocity, *m* is the inner circle limit and *eps* is the convergence limit. SA is based on the well-known Metropolis Monte Carlo acceptance criterion, which models how an ensemble of particles moves, in a quasi-ergodic way, from its current thermodynamic state to a minimum energy final state. For this reason, the candidate solutions will be accepted with a Boltzmann-like probability function (consider the *if* test in Figure 3). As the temperature decreases, less efficient moves occur less frequently and the solution converges to an upper bound rather than to the optimal solution.
Now, the SA code can be easily applied to a telescope scheduling problem. Considering, for instance, a possible SRT schedule composed by 48 temporal slots (epochs) and each one taking 1 hour, the code generates a new solution (*fn*) at each iteration by "perturbing" the current solution (*f*). At each iteration is guaranteed the compatibly with the current timetable and, hence, with the slots statically allocated to the constrained activities (planned maintenance, VLBI runs and ToOs).

```
Select an initial solution f(x1,..., xN)
Set T=1000, a=0.85, m=100, eps=1E-5
Repeat,
   Set i=0
   Repeat,
       Repeat,
         Select randomly j with xj=0 and obtain a new fn
       Until fn is feasible
       Set D=f-fn
       If D>0 OR RND(0,1)< exp(D/T)
         Set f=fn
       End If
       Set i=i+1
   Until i=m
   Set T=T*a
Until T<eps
```

Fig. 3 Pseudo-code implementing the SA algorithm. The objective function *f*, defined in Equation 1, is even the KP solution once the pseudo-code converges. The feasibility of a new solution *fn* depends on the requirements and constraints inherent each specific optimization problem.

Table 1 and 2 depict a 48-hour schedule, where, for simplicity, only a sub-set of 12 slots (grey-shaded rows), enumerated from 13 to 24, is shown. The original set (Table 1) is perturbed selecting the slot 16 and replacing the observation #7 with the observations #13 and #3 (Table 2). Since observation #13 takes 3 slots (from 16 to 18, see Table 2) even the observation #11 has to be replaced and new slots (from 19 to 20) are available for allocating the observation #10 taking two slots (see again Table 2). If the new schedule turned to be better than the previous one, then the observations #7 and #11 will be added to the other observations waiting for a new allocation; otherwise the previous schedule will be restored and a new iteration occurs.

13	14	15	16	17	18	19	20	21	22	23	24
#8	#7	#7	#7	#11	#11	#11	#11	#16	#16	#24	#24

Tab. 1 Example of a 12 slot sub-set (from epoch 13 to 24) of a 48-slot schedule during the annealing process. Slots are indicated in grey-shaded rows. Observations are indicated with a '#' sign.

13	14	15	16	17	18	19	20	21	22	23	24
#8	#3	#3	#13	#13	#13	#10	#10	#16	#16	#24	#24

Tab. 2 New schedule after perturbing the slot 16 by replacing the observation #7. This perturbation causes different replacements: the observation #7 with the observations #3 (in the slots 14 and 15) and #13 (in the slots 16) and the observation #11 with the observation #13 (in the slots from 17 to 18) and #10 (in the slots from 19 to 20).

4 Simulation results

In order to check the SA algorithm in a real DS problem, for instance the scientific observation allocation at a telescope, a 48-hour long schedule (the extension of our WFM forecast), composed by 30 observations and jobs (fixed and not-fixed activities, all gathered in a suitable pool) which can take from 2 to 6 hours, is here considered. Each not-fixed activity gets a score (the scientific worth or priority), while the fixed ones get no score because are constrained to a specific epoch. Moreover, a subset of observations, whose allocation might depend on the environmental conditions, gets an additional time-dependent score related to the atmosphere status forecast. The only topological constraint applied to the SA algorithm implementation was to avoid overlapping of observations or jobs.
The SA code was run 25 times with the same candidate set in order to check the statistical consistency of the solutions. We obtained 25 statistically equivalent solutions having (about) the same final score ($\mu = 101.1$, $\sigma = 2.5$), but with different tailings. SA does not necessarily provide with the optimal solution, instead it simply converges asymptotically to a *good enough* solution.
The asymptotic behavior of one of the 25 simulations is shown in Figure 4. At the beginning, when the temperature is high all the moves are accepted (in Figure 4 the green curve approaches 100%). In this early stage, the schedule still has not optimized, but it is topologically consistent because all constraints have been already fulfilled. As the temperature decreases the number of accepted moves decreases (in Figure 4 the green curve starts approaching 0% after about 10 s). Then, the convergence begins and the solution freezes to the final configuration.

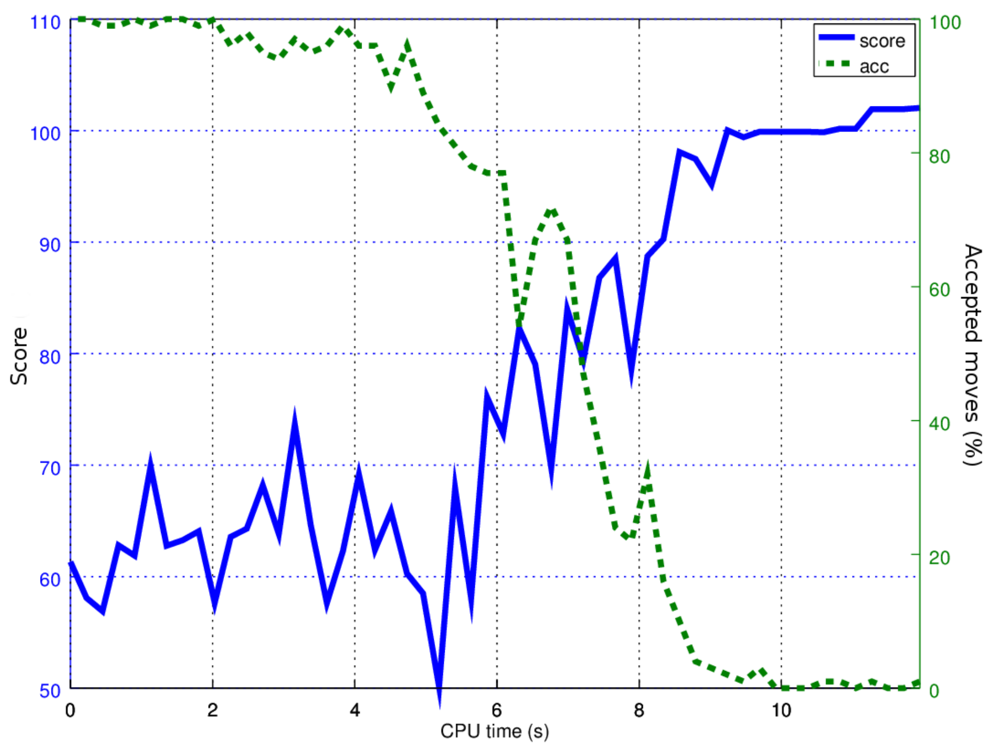

Fig. 4 Total score (blue trace) and percentage of accepted moves (green trace) during a simulation taking about 12 s of CPU time. During the annealing (while temperature decreases) the total score increases from 60 to 102 thanks to the optimization process.

Table 3 shows the result of the SA simulation after optimizing a 48-hour schedule. This is one of the 25 obtained solutions allocating not-fixed observations (black code, see for instance observations #9 in the slots from 1 to 4 and #2 in the slots from 9 to 11), fixed observations (red code, see for instance observations #12 in the slots from 5 to 8 and #13 in the slots from 15 to 16) and weather dependent observations (green code, see for instance observations #5 in the slots from 13 to 14 and #30 in the slots from 17 to 19). This solution fulfills all the requirements and constraints and leaves a 1-hour not-allocated slot (see slot 12), which, however, can be easily allocated in the aftermath by the human supervision.

1	2	3	4	5	6	7	8	9	10	11	12
#9	#9	#9	#9	#12	#12	#12	#12	#2	#2	#2	#0
13	14	15	16	17	18	19	20	21	22	23	24
#5	#5	#13	#13	#30	#30	#30	#21	#21	#21	#21	#21
25	26	27	28	29	30	31	32	33	34	35	36
#11	#11	#11	#11	#11	#11	#4	#4	#4	#14	#14	#29
37	38	39	40	41	42	43	44	45	46	47	48
#29	#10	#10	#8	#8	#24	#24	#24	#15	#15	#15	#25

Tab. 3 An optimized 48-hour schedule resulting by a SA simulation. Each slot is allocated to not-fixed observations (labelled by black code), fixed observations (red code) and weather dependent observations (green code). Epoch 12 is not allocated by the algorithm and will be assigned to observation #2 or #5. Observation #25 will be shared with the next schedule. Again, slots are indicated in grey-shaded rows and observations with a '#' sign.

5 Discussion and future developments

This paper is a first attempt at facing the possibility to implement a DS for the SRT. After considering the DS approaches at some of the most important astronomical facilities, a heuristic approach to the DS optimization defined in terms of knapsack problem has been proposed. Though our results are not conclusive, they do suggest that a good optimization degree is achievable even taking into account constrained (fixed) activities.

In the SRT early science phase, the antenna overall scheduling time was relatively, although understandably, rigid. Conversely, a certain degree of freedom in the duration of the observations and of the maintenance activities is desirable, as shown in the Section 4, where time slots from 2 to 6 hours were considered. The inclusion of short time-slots may increase the allocation efficiency and more in general the schedule effectiveness, although they may be more difficult to be managed from an operative point of view.

A weather-dependent score is also desirable to improve the telescope scientific results mainly at the high frequency. By using the DS, this kind of score would be easily implemented in the observation allocation and related to those predictable quantities (opacity, integrated water vapor, system temperature, wind and rain) impacting on the antenna efficiency, pointing precision and safety.

Looking ahead, additional parameters such as the minimization of observation fragmentation, the object observability and the technical overhead to set a specific observation (e.g. cable unwrapping, receiver changing) could be considered to get a more reliable score.

Finally, it is worth noting that many of the information contributing to define the DS scores (scientific scores apart) may be gathered from a proposal submission system (PSS). In fact, a PPS, designed to evaluate (or infer) many technical parameter related to the observations (source visibility, air masses, required Tsys threshold, sensitivity, scheduling constraints and so on), should be integrated in a framework together with the scheduler system proposed in this report.

References

1. Hartman, J.C., Engineering Economy and the Decision-Making Process, Prentice Hall, New Jersey, 2006

2. Feldmann, A., Sgall, J., Teng, S., Dynamic scheduling on parallel machines, Theoretical Computer Science, 130, 1994

3. Mora, M. and Solar, M., A survey on the dynamic scheduling problem in astronomical observations, Artificial Intelligence in Theory and Practice III, Springer, IFIP Advances in Information and Communication Technology, 2010

4. Fraser, S.N., Adaptive optimal telescope scheduling, PhD Thesis, Liverpool John Moores University, 2012

5. Farris, A. and Roberts, S., Dynamic Scheduling in ALMA, Astronomical Data Analysis Software and Systems XIII, ASP Conference Series, 314, 2004

6. Buchner, J., Dynamic Scheduling and Planning parallel observations on large Radio Telescope Arrays with the Square Kilometre Array in mind, Auckland University of Technology, 2011

7. Sessoms, E., Clark, M. H., Marganian, P., McCarty, M. and Shelton, A., The GBT Dynamic Scheduling System: Scheduling Applications of the Knapsack Problem and Sudoku, Astronomical Data Analysis Software and Systems XVIII, ASP Conference Series, 411, 2009

8. Yamada, T., Nakano, R., Job-shop scheduling, Genetic algorithms in engineering systems, IEE control engineering series, 1997

9. Clark, M. H.; Balser, D. S., Braatz, J., Condon, J., Creager, R. E., McCarty, M. T., Maddalena, R. J., Marganian, P., O'Neil, K., Sessoms, E., Shelton, A. L., The GBT Dynamic Scheduling System: An Update, Astronomical Data Analysis Software and Systems XX, ASP Conference Proceedings, 442, 2011

10. Buffa, F., Bolli, P., Sanna, G. and Serra, G., “An atmosphere monitoring system for the Sardinia Radio Telescope", Measurement Science and Technology, 28, 2017

11. Buffa, F., Serra, G., Bolli, P. et al., “K-band system temperature forecast for the Sardinia Radio Telescope”, OAC Internal Report N. 54, 2016, https://goo.gl/qbPJl9

12. Martello, S., Toth, P., Knapsack Problem, Algorithms and Computer Implementations, John Wiley & Sons, 1990