

Optimization of telescope scheduling

Algorithmic research and scientific policy

A. I. Gómez de Castro¹ and J. Yáñez²

¹ Instituto de Astronomía y Geodesia (CSIC-UCM), Universidad Complutense de Madrid, 28040 Madrid, Spain

² Department of Statistics and Operations Research, Universidad Complutense de Madrid, 28040 Madrid, Spain
e-mail: jayage@mat.ucm.es

Received 13 February 2002 / Accepted 28 October 2002

Abstract. The use of very expensive facilities in Modern Astronomy has demonstrated the importance of automatic modes in the operation of large telescopes. As a consequence, several mathematical tools have been applied and developed to solve the (\mathcal{NP} – *hard*) scheduling optimization problem: from simple heuristics to the more complex genetic algorithms or neural networks. In this work, the basic scheduling problem is translated into mathematical language and two main methods are used to solve it: neighborhood search methods and genetic algorithms; both of them are analysed. It is shown that the algorithms are sensitive to the scientific policy by means of the definition of the objective function (F) and also by the assignment of scientific priorities to the projects. The definition of F is not trivial and requires a detailed discussion among the Astronomical Community.

Key words. methods: observational

1. Introduction

The use of very expensive facilities in Modern Astronomy has demonstrated the importance of automatic modes in the operation of large telescopes. As a consequence, several mathematical tools have been applied and developed to solve the (\mathcal{NP} – *hard*)¹ scheduling problem: from simple heuristics to the more complex genetic algorithms or neural networks. Interesting applications of new mathematical tools and artificial intelligence to astronomical issues may be found, for instance, on the JPL web page² or in some conference proceedings as e.g. Dasgupta & Michalewicz (1997).

Needless to say, when a Committee for Assignment of Time (CAT) decides that a given project *must* be carried out, this is indeed carried out, unless the observing conditions are so critical that, for instance, an unexpected change in the weather

conditions makes it unfeasible. However, this is not the general case. Most of the observations enter into a general pool of good projects that are scheduled in order to satisfy the astronomical constraints (e.g. the observability of the target) and the instrumental constraints (e.g. the availability of a given instrument or the calibration sequences). The total exposure time available is well-known in space projects thus, optimization implies solving a complex scheduling problem with hard and soft constraints; its classical example is the programme based on the use of neural networks that was developed for the Hubble Space Telescope (Johnston & Adorf 1994). The programme is by now widely used in the community and modified versions have been applied to other telescopes, including ground based telescopes, such as the VLT (Giannone et al. 2000). Recent developments in the application of automated modes to telescope scheduling and operation (from small robotic telescopes to telescopes larger than the Earth, such as the VSOP-VLBI Mission) are summarized in the proceedings of some recent conferences (see e.g. Ford 1998; Manset et al. 2000) to mention but a few recent ones.

Unfortunately, the amount of time available for observations is not known a priori for ground-based telescopes, especially in the infrared. Therefore, CATs usually assign more observing time than is actually available. In the traditional observing scheduling, this is translated into having a good or bad observing run and, maybe, having to resubmit the scientific

Send offprint requests to: A. I. Gómez de Castro,
e-mail: aig@mat.ucm.es

¹ The scheduling problem has an inherent computational complexity: the number of solutions grows exponentially with the number of targets and there is no algorithm that can avoid the explicit or implicit enumeration of all the solutions. The computational complexity is a rigorous mathematical discipline that shows how most of the optimization problems can be grouped into classes such that all of the problems in the same class are of similar complexity; the \mathcal{NP} – *hard* problems are the most important class.

² URL: www-aig.jpl.nasa.gov

project for the next year. In this sense, the advent of automatic scheduling implies a substantial improvement since whenever the CAT has established that a project is good enough, automatic scheduling will try to carry it out until the very last feasible date of the semester. Therefore, automatic scheduling allows taking full advantage of the variations of the weather conditions. Moreover, automatic scheduling also allows defining calibration sequences in a uniform manner and automatic scheduling supports the creation of Uniform Data Archives for ground telescopes. These type of Archives are available already for several space missions (IRAS, IUE, ROSAT, HST, ISO...) and they have proven to be a very useful tool for scientific research; in fact, there is a growing community of astronomers carrying out research based on Archive Data.

However, automatic scheduling requires the definition of mathematical functions describing the optimal performance of the facility. In principle, this definition looks rather simple: scientific excellence (carrying out the maximum of scientific programs of the highest scientific quality) with the minimum cost or in the most effective manner (minimizing instrument and calibration overheads, taking profit of the changing weather conditions). However, the mathematical definition of this function determines the result (the final scheduling of the observations). An additional issue that must be taken into account is the classification of the scientific quality of the proposals by the CAT; the translation of this classification into mathematical language may also be relevant. Finally, the solution of (\mathcal{NP} – *hard*) scheduling problem depends on the algorithm used since it is an heuristic proposed to find rapid solutions close to the true optimal solution. In this work, these issues are analysed placing special emphasis on the use of genetic algorithms. Other metaheuristics can be applied to solve these scheduling problems as neural networks (see e.g. Martínez et al. 2002).

In May 1998, a study panel was funded on “An Integrated approach to Telescope Operations and Scheduling” under the auspices of the European Commission, through the Training and Mobility of Researchers Programme of the Forth Framework Programme for Research and Technology Development. The panel examined several aspects related to the operation of large astronomical facilities in Europe. An important point was the analysis of the best procedures for telescope scheduling, in order to optimise the science output from the telescope/observatory³. A final recommendation of the Panel was to make the astronomical community aware of some implications of the automatic scheduling. This article is also intended to satisfy this recommendation.

The work is structured as follows. The scheduling problem is translated into mathematical language in Sect. 2. Two main methods are used to solve it: neighborhood search methods and genetic algorithms. In Sect. 3, these methods are applied to some test samples and the main results (computer performance, degree of optimization) are analysed. In Sect. 4, the role of some possible scientific policies in the final scheduled observations is analysed. The main results are summarized in Sect. 5. A detailed description of the algorithms and their

performance, as well as some technical aspects are described in Appendices A to C at the end of the article.

2. The mathematical formulation of the basic scheduling problem

The general scheduling problem consists of determining the most efficient procedure for the observation of a set of astronomical targets with a given telescope. Each target is qualified with five main parameters: the astronomical coordinates (e.g. right ascension and declination), the exposure time, the instrumental set-up and the scientific quality of the project. In addition, there is a temporal constraint (observability) that can be defined either by the observatory (e.g. visibility of the target) or by the observer (simultaneous observations with other observatories, monitorings etc.). Finally, it must be taken into account that the peer refereeing of scientific proposals involves the qualification of projects (instead of targets) so, an additional constraint must be added to ensure that all the targets of a given project are treated equally, independent of their temporal constraints.

Scheduling can be broken down into long term and short term components or “granularities”:

- The *long-term* component deals with the scheduling of the observations over a time span of roughly a year. Observing constraints (e.g. sun, moon, observer-specified or site-specific constraints...) must be handled; as a consequence, the scheduling problem becomes a sophisticated search of the best feasible solution in the most (cost and time) efficient manner. As the site-specific constraints are very important for the long-term component, the optimal solution of the scheduling problem depends strongly on the astronomical facility and, for this reason, it is difficult to define a common, abstract space where different algorithms and techniques are tested and compared. In fact, there are active research programs engaged in defining a common modeling language for planning and scheduling (see e.g. URL: www.planning.systems.org/index.html). An additional property of long term scheduling is that it is not carried out in real time and therefore the use of very time-efficient algorithms it is not as critical as in short-term scheduling⁴.
- The *short-term* component deals with the scheduling of the observations over 1–3 days. Some of the observing nights scheduled by the long-term component may be lost because of bad weather conditions, failures in the telescope or instrument, or because of the observation of targets of opportunity. This overflow of unexpected activities requires reworking the scheduled queues but now with an oversubscription factor that needs to be absorbed as soon as possible because of the observability constraints (or the

³ URL: <http://www.iac.es/otri/panel/tpanel.htm>

⁴ As an example, we have developed a program for managing the scheduling of the 10-m GTC telescope in La Palma (Spain) that, is based on a greedy strategy so that the optimal observing dates for each target are calculated and the observations are scheduled accordingly; genetic algorithms are used to refine the scheduling for regions of the sky that may be oversubscribed.

availability of a given instrument)⁵. In this context, the problem can be simplified to the scheduling of a list of observations with several possible instrumental configurations and without observing constraints. Therefore, the short-term component is basically independent of the astronomical facility and provides a good observatory-independent arena to create a mathematical model and perform experiments.

The main objective of this work is to analyze the sensitivity of the scheduling process with respect to the scientific policy, e.g. the definition of the objective function and the assignment of scientific priorities. As a natural consequence, short-term scheduling is the most appropriate for these tests.

To fulfil the objectives of this work, we have defined a *basic scheduling problem* to run our tests; following Drummond et al. (1994), we shall consider N astronomical targets to be scheduled. Each target $j \in \{1, 2, \dots, N\}$ is characterized by:

1. A priority $q(j) \in Q = \{1, 2, \dots, q\}$, which is assigned by the CAT following an anticipated scientific return. This parameter reflects the scientific relevance of the observation; the more important, the smaller $q(j)$ is assigned. The set Q is linearly ordered and can be supposed without any loss of generality that the maximum allowed value is $q = 4$.
2. The right ascension $\alpha(j)$ and declination $\delta(j)$ of the target j
3. The time required to carry out the observation, $\tau(j)$, which includes the time required for the instrumental set-up and calibration, as well as the exposure time.

Given a horizon time T (or total available time), several scheduling problems can be stated depending on the assumptions of the model and depending also on the criteria of the telescope manager.

Let $f(q)$ be a weighting factor associated with each priority level $q \in Q$ verifying,

$$f(r) \geq f(s) \quad \forall r, s \in Q \text{ with } r \leq s.$$

A basic scheduling problem is stated when the objective is to maximize the total weight of the targets scheduled along the horizon time T . The objective function to maximize is

$$F = \sum_{i=1}^{N_s} f(q(S(i)))$$

where $S(i)$ is the i th target scheduled and N_s is the total number of targets scheduled, with $N_s \leq N$.

This basic scheduling problem can be stated as a permutation problem in the sense that the optimal solution can be characterized as a permutation $(S(1), \dots, S(N_s))$ of the targets. Such a permutation identifies the order in which the targets must be consecutively performed, i.e. target $S(i+1)$ follows to target $S(i)$. Consequently, the decision variable will only consider the order in which the targets are observed.

⁵ Also the design of the telescope may be optimized, including back-up instruments that are not very demanding of the weather conditions. Returning to the example of the 10-m GTC, the prime instrument may be an infrared spectrograph and an optical spectrograph may be used as a back-up instrument. As a result, it should be feasible to re-schedule the observations even in real-time.

This problem is classified as an \mathcal{NP} – *hard* problem, see Garey & Johnson (1979). The computation time grows exponentially with the parameter N and only approximate algorithms or heuristics can be used to solve moderate and large size problems.

In this way, the decision variable will be the number of scheduled targets N_s and their order, which will be characterized by the vector

$$S = (S(1), S(2), \dots, S(N_s))$$

with $S(i) \in \{1, \dots, N\} \forall i \in \{1, \dots, N_s\}$.

Any vector S will be a valid solution if it fulfills the time requirements along the horizon time T .

Let $[t_i^1, t_i^2]$ be the time interval assigned to target $S(i) \in \{1, \dots, N\}$ belonging to the valid solution. So that the arrangement S characterizes a valid solution, the following conditions must be fulfilled:

1. $t_i^2 \geq t_i^1 + \tau(i) + t_{\text{ini}}(i)$
 $\forall i \in \{1, \dots, N_s\}$ where $t_{\text{ini}}(i)$ includes the telescope and instrument commands time required by the scheduled target $S(i)$ taking into account the previous target $S(i-1)$.
2. $t_1^1 \geq 0$
3. $t_i^1 \geq t_{i-1}^2 \quad \forall i \in \{2, \dots, N_s\}$
4. $t_{N_s}^2 \leq T$.

Given a horizon time, T , and an arrangement of the N targets

$$\sigma = (\sigma(1), \sigma(2), \dots, \sigma(N))$$

a valid solution for the basic scheduling problem, characterized by the N_s scheduled observations

$$S = (S(1), S(2), \dots, S(N_s))$$

with objective function F , is constructed with a greedy procedure. Notice that N is the total number of targets and N_s will be the number of targets that can be scheduled in the horizon time T .

As a permutation problem, every arrangement of observations σ contains a feasible scheduling S ; the optimization problem looks for the optimum solution among them. Taking into account that the solution of the optimization problem is associated with an optimum arrangement σ^* , the permutation problem is focused to the search for this vector σ^* . Other authors do not constrain the problem to feasible solutions, allowing temporal or other constraint violations, and eliminating them afterwards so that the schedule is feasible; see, for example, Johnston & Miller (1993).

3. Computational experiences

Two general heuristic procedures will be used: neighborhood search methods (see Appendix A) and genetic algorithms (see Appendix B). We have studied their generic performance, basically, computer performance and degree of optimization achieved. We have simplified the astronomical scheduling problem to the scheduling of the observations of N targets

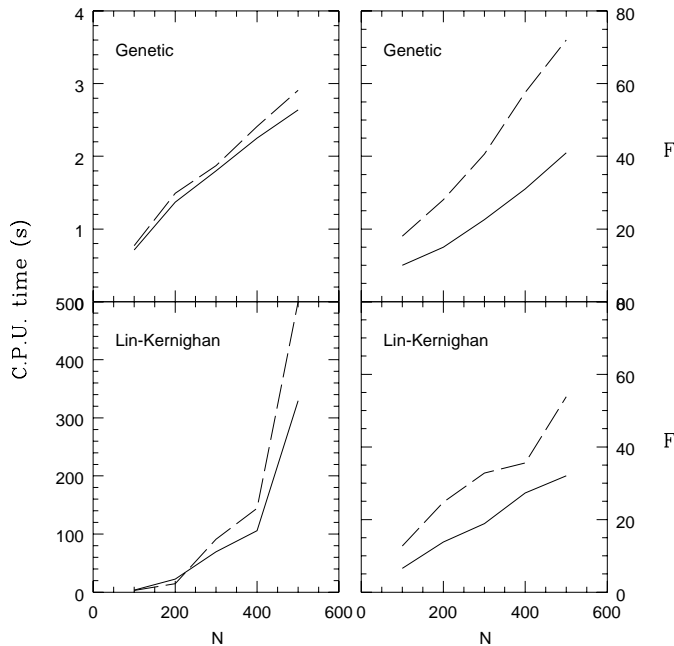


Fig. 1. Scaling of the computer performance with the number of observations, N for the Lin-Kernighan and the genetic algorithms. The C.P.U. time and the objective function F are shown in the left and right panels respectively. The results are shown for two oversubscription factors: 1.5 (dashed) and 3 (solid).

with four possible instrumental configurations and without observing constraints (Sun, Moon, telescope dead zones, Earth shadow, etc.). The maneuvering time has been parameterized by a constant angular velocity. The overheads caused by the change of the instrumental configuration are assumed to be short, in between 2 and 6 min depending on the instrument. Notice that if the instrumental overheads were comparable to or substantially longer than the typical exposure time, a *queue-scheduling* by instruments will be required to optimize the usage of the telescope time but then, again, we should solve the scheduling problem for each queue. The test have been run on a Pentium III 450 MHz.

As a first step, we checked the dependence of the computer performance on the number of targets, N , randomly distributed in $\pi^2/2$ rad² in the sky. As shown in Fig. 1, as N increases, the performance of genetic algorithms scales much better than the performance of the Lin-Kernighan algorithm, a neighborhood search method (see Appendix A).

However, the distribution of the astronomical targets in the sky is not random at all (e.g. the stellar targets are concentrated in the galactic plane). To take this fact into account, we have run some experiments with a target list that has been extracted from the original *Infrared Space Observatory* (ISO) targets list. Coordinates, instruments and exposure times have been preserved; targets with exposure times longer than 4500 s have been deleted from the list.

Two data files have been selected:

1. An *exponential distribution* test list. 140 observations have been selected at random from the original ISO list. The histogram of the distribution of the observations by exposure times is exponential-like (see Fig. 2).

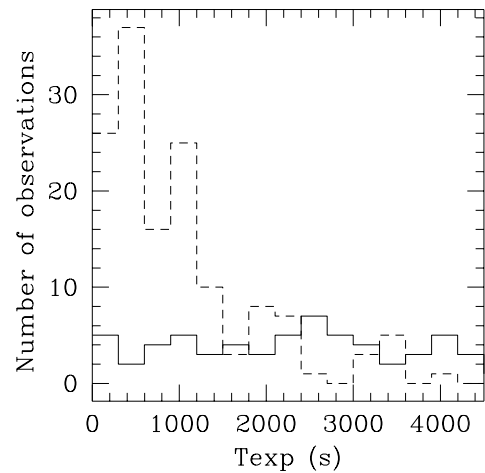


Fig. 2. Histograms displaying the distribution function with the exposure time of the input scheduling list for the tests: exponential (dashed) and flat (solid) input list.

2. A *flat distribution* test list has been obtained after some manipulation of the original ISO file (see Fig. 2); only 60 targets are included.

To complete the information in the lists, priorities q have been assigned to each target. Two different policies have been followed:

1. A *grey policy* that assigns $q = 1, 2, 3, 4$ to each quarter of the targets in the list so, the probability of finding a target in the list with $q = 1$ is the same than the probability of finding a target with $q = 4$.
2. A *standard policy* that assigns $q = 1, 2, 3$ and 4 to the 10%, 60%, 20% and 10% of the targets so, most of the targets have $q = 2$.

In both cases the values of q have been assigned randomly to the targets in the lists. In summary, four input samples have been generated to test the properties of the heuristics described above: *exponential* or *flat* distributions with *grey* or *standard* policies.

The programming of the neighborhood search methods (see Appendix A) is direct and the computation is stopped after convergence (when further iterations do not improve the objective function F). However, the programming of the genetic algorithms required a finer tuning. The genetic algorithm (see Appendix B) works well with the following parameters:

$$n = 50; it_M = 10; p_c = 0.6; p_m = 0.1;$$

the computations stop when the maximum number of iterations is achieved.

The main numerical results are summarized in Appendix C for oversubscription factors 1.5 and 3 (see also Fig. 3). These factors have been set up so the horizon time T is $2/3$ and $1/3$ of the total exposure time $\sum_{i=1}^N \tau(i)$ (see Sect. 2). In fact, the real oversubscription factors are ~ 2 and ~ 4 due to the time consumed by maneuvering the telescope and setting up the instruments.

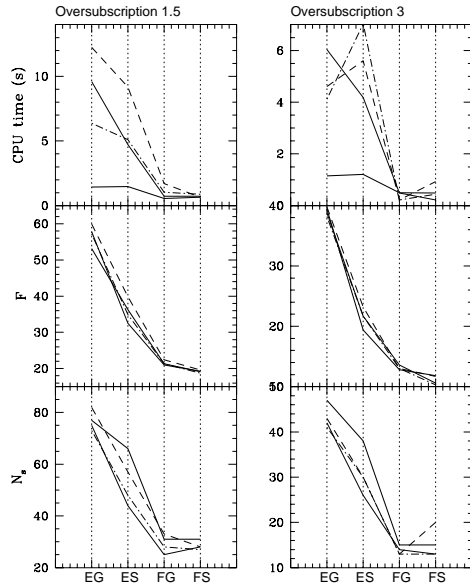


Fig. 3. Summary of the computational experiences (see also Tables C.1 and C.2). The results for three neighborhood search methods (simple, 2-opt and Lin-Kernighan) as well as for the genetic algorithms (with cross-over operator C1) are plotted with thin, dot-dashed, dashed and thick lines respectively. EG, ES, FG and FS stand for data files with exponential (E) or flat (F) distributions and grey (G) or standard (S) policies.

3.1. Scientific optimization

The value of the objective function F and the number of targets of the initial set observed depends on the following factors:

1. *Policy*: A “grey” policy, with the same fraction of observations in every category, produces better results than the so-called “standard” policy. Also the percentile of first class projects ($q = 1$) scheduled is larger.
2. *Algorithm*: The best results (higher values of F) are achieved with the Lin-Kernighan algorithm whenever the list of targets is pre-ordered by right ascension. Notice that however, this is done at the cost of more CPU time. Also two crossover operators have been tried in genetic algorithms; algorithm C1 seems to produce slightly better results (see the technical appendix for more details).
3. *Oversubscription factor*: High oversubscriptions force the selection of the best qualified projects.

In summary, the best results (the largest values of F and N_s) are generally achieved with a simple neighborhood search method such as the Lin-Kernighan heuristic whenever the sample is pre-ordered by right ascension. This is because the neighborhood search methods are efficient to search for the local maximum of F closest to the initial solution; this, in turn, depends strongly on the initialization procedure being very good if the initialization is able to take advantage of an a priori good knowledge of the problem. The genetic algorithms are clearly superior if there is not a good knowledge of the main constraints in the scheduling of the observations. Notice that although they do not take advantage of an a priori knowledge of the system (the two cross-over operators that we have tried for this purpose do not seem to have been very successful),

Table 1. Scientific policies.

$f(q)$	$q = 1$	$q = 2$	$q = 3$	$q = 4$
High pass-band (H)	1	0.4	0.3	0.1
Step (S)	1	0.9	0.2	0.1
Low pass-band (L)	1	0.8	0.7	0.1

they are able to find very good solutions close to the maximum and much more rapidly than the neighborhood search methods.

Finally, we want to remark that none of these heuristics guarantee that all the best projects are carried out, in fact, a significant fraction of observations with $q > 1$ are accommodated in the final scheduled observations in order to maximize F .

4. The role of the scientific policy

We have run some further tests to get further insight into the sensitivity of the algorithms to the scientific policy. Only the best-behaved algorithms: Lin-Kernighan (with an initial sample ordered by right ascension) and genetic algorithms have been applied to the tests. We have selected only those test samples with grey policies and we have defined three different sets of weighting factors (see Table 1).

The three sets mimic three basic policies: *step*, *low pass-band* and *high pass-band* that may (or may not) reinforce the selection of high quality projects.

Only simulations with an horizon time T corresponding to an oversubscription factor of 1.5 have been carried out. The results are summarized in Table 2 where the total number of scheduled projects with $q = 1$, $q = 2$, $q = 3$ and $q = 4$ is given. The percent of projects with $q = 1 \dots 4$ that is finally scheduled, is illustrated in Fig 3. The main conclusions that can be drawn are:

1. It is difficult to reinforce high quality observations by tuning up the scientific policy.
2. The final results depend strongly on the characteristics of the sample, especially on the availability of good observations ($q = 1$) with short exposure times.
3. The Lin-Kernighan algorithm responds in the most desirable manner to the scientific policies for this sample.

5. Summary and conclusions

We can summarize the results of this research in the following points:

1. The neighborhood search algorithms and the genetic algorithms deal with the optimization problem in different manners and this shows in the results. For large size problems, genetic algorithms are by far the best. However, for small size problems, neighborhood search algorithms are efficient to search for the local maximum of F closest to the initial solution; this property depends strongly on the *goodness* of the initialization procedure, e.g. the ability to take advantage of an a priori good knowledge of the problem. The genetic algorithms are clearly superior if there is not a

Table 2. The role of the scientific policies (oversubscription factor 1.5).

Input sample ^a	Algorithm ^b	Policy ^c	F	N_s	$N_s(1)$	$N_s(2)$	$N_s(3)$	$N_s(4)$
Exponential	L-K (Ordered)	H	51.6	80	35	31	14	0
		S	68.6	77	35	35	6	0
		L	66.6	75	35	35	4	0
	Genetic – C1	H	46.7	67	35	27	2	3
		S	60.0	67	35	27	2	3
		L	64.1	75	35	23	15	2
	Genetic – C2	H	47.2	72	35	27	2	8
		S	61.7	68	35	29	2	2
		L	60.1	73	34	30	2	7
Flat	L-K (Ordered)	H	20.4	32	15	7	8	2
		S	24.2	26	15	10	1	0
		L	24.6	31	14	5	9	3
	Genetic – C1	H	19.0	28	14	8	6	0
		S	21.9	26	14	8	3	1
		L	24.8	30	12	8	9	1
	Genetic – C2	H	20.3	33	14	8	10	1
		S	23.0	27	15	8	4	0
		L	23.4	30	6	12	11	1

^a Input sample indicates the distribution function of the exposure times in the input list (exponential or flat)

^b Algorithm indicates the heuristic used: the Lin-Kernigan algorithm (with targets pre-ordered by right ascension) or genetic algorithms with cross-over operators C1 or C2 (see Appendix B).

^c Identifies the scientific policy: high pass-band (H), step (S) and low pass-band (L).

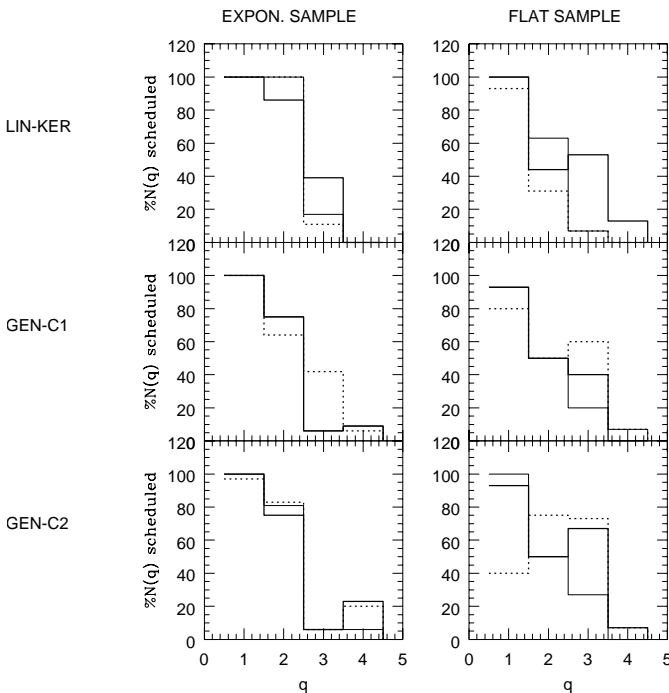


Fig. 4. Effect of the scientific policies in the finally scheduled observations. The three policies used for the simulation are *high passband*, *step* and *low passband* and the corresponding results are marked with thick, thin and dotted lines respectively.

good knowledge of the main constraints in the scheduling of the observations. Moreover, genetic algorithms are rapid enough to allow a fast re-organization of the scheduling

queues that can be as rapid as the variations of the weather conditions (few seconds).

2. To take advantage of the automatic scheduling, it is important to have a small degree of oversubscription in the final allocated time and a large number of observations requiring short exposure times for instance, the SNAPSHOT proposals for the Hubble Space Telescope.
3. The algorithms are sensitive to the scientific policy by means of the definition of the function F and also by the assignment of priorities to the projects. The definition of F is not trivial and requires a detailed discussion among the Astronomical Community. Notice that much more complex functions taking into account the number of targets per observing programme or the number of observing modes or the volume of information contained in the data (e.g. spectral range or the spatial size) can be thought of. Also the way in which the CATs qualify the observing proposals is relevant. Currently, CATs qualify the proposals in very different manners depending on the European Facility.
4. The only way to ensure that first class projects are carried out is by introducing directly constraints in the scheduling programme since the heuristics do not guarantee that they are scheduled instead of more convenient projects.

As a final remark, we think that it would be very interesting to make a comparative analysis between traditional observations and automatic scheduling for the Large European Telescopes. This study would allow one to test the scheduling algorithms in a “real” framework and show clearly the advantages of automatic scheduling. This will also allow us to make a more

realistic study where the role of some observing constraints such as air mass and temporal links between observations could be taken into account.

Acknowledgements. We thank Daniel Ponz for his critical reading of the text. We also would like to thank the referees of this article: Drs. M. Giuliano and G. Miller for their suggestions.

Appendix A: Neighborhood search methods (NSMs)

The procedure to search for an optimal solution with these methods is as follows. The first arrangement of observation σ^0 is constructed with an initialization procedure. This vector σ^0 is modified in successive iterations to improve the objective function. Depending on the improvement heuristic, several algorithms can be designed. All of them are obtained by changing two or more elements of the previous permutation.

Depending on the initialization procedure and on the definition of the set of neighbors, different algorithms are defined.

A.1. Initialization procedures

A.1.1. Random arrangement

For the basic scheduling problem, a simple initialization will be the identity arrangement

$$\sigma^0(j) = j \quad \forall j \in \{1, \dots, N\}$$

In this arrangement, there is not any particular pre-ordination of the N projects. This procedure is very simple and general for any optimization problem, but it does not take advantage of some well known constraints in astronomical problems.

A.1.2. Ordered arrangement

Any optimized scheduling will try to minimize the maneuvering time and the overheads caused by the instrumental set-up and the calibration sequences. Henceforth, the sorting of the data by instruments and modes and also by astronomical coordinates assists the optimization procedure in NSMs. It is well known that neighborhood search algorithms explore the space of solutions σ near the initial solution σ^0 and has a strong tendency to generate “optimal solutions” which are only the local minimum (Rayward-Smith et al. 1996): these heuristics have difficulties in exploring farther out the local minimum of the surface F in the space of feasible solutions that grows exponentially with N . Therefore, only if the initial solution σ^0 is close to the absolute minimum will the optimization be really optimal.

To test the relevance of this sorting, the initial arrangement has been pre-ordered by right ascension with the following method:

1. Start with an arbitrary observation $\sigma^0(1)$.
2. Given the current partial arrangement $\sigma^0(1), \sigma^0(2), \dots, \sigma^0(k)$, the next observation $\sigma^0(k+1)$ will be that whose right ascension $asc(\sigma^0(k+1))$ is the closest to $asc(\sigma^0(k))$ and that it is not already included in the arrangement.

3. Halt when the current arrangement contains all the observations.

A.2. Improvement procedures

Given an arrangement, σ , a new permutation σ' is derived:

$$\sigma' = (\sigma(1), \dots, \sigma(j_1 - 1), \sigma(j_1), \sigma(j_1 + 1), \dots, \sigma(j_2 - 1), \sigma(j_2), \sigma(j_2 + 1), \dots, \sigma(N)) \quad (\text{A.1})$$

If the objective function, F , is improved after this permutation, then σ is updated by σ' . Different algorithms have been defined for the permutation. In this article we describe but a few, namely, the simple improvement procedure, the 2-opt exchange improvement and the Lin-Kernighan algorithm. These three algorithms have been used to solve a particular permutation problem: The Traveling Salesman Problem (TSP), see Ball et al. (1995).

A.2.1. Simple improvement

The new permutation σ' is derived changing the elements $\sigma(j_1)$ and $\sigma(j_2)$:

$$\sigma' = (\sigma(1), \dots, \sigma(j_1 - 1), \sigma(j_2), \sigma(j_1 + 1), \dots, \sigma(j_2 - 1), \sigma(j_1), \sigma(j_2 + 1), \dots, \sigma(N)) \quad (\text{A.2})$$

In this heuristic, the process continues until any couple $j_1, j_2 \in \{1, \dots, N\}$ does not improve the actual permutation.

A.2.2. 2-opt exchange improvement

In the previous heuristic, two elements of the permutation are interchanged, the other elements are in the same position. In this heuristic, however, given two elements, the second is placed after the first one, and all intermediate elements are delayed one position.

$$\sigma' = (\sigma(1), \dots, \sigma(j_1 - 1), \sigma(j_1), \sigma(j_2), \sigma(j_1 + 1), \dots, \sigma(j_2 - 1), \sigma(j_2 + 1), \dots, \sigma(N)) \quad (\text{A.3})$$

A.2.3. Lin-Kernighan algorithm

In this heuristic, some major changes are introduced in the permutation σ' .

$$\sigma' = (\sigma(1), \dots, \sigma(j_1 - 1), \sigma(j_2), \sigma(j_2 - 1), \dots, \sigma(j_1 + 1), \sigma(j_1), \sigma(j_2 + 1), \dots, \sigma(N)) \quad (\text{A.4})$$

Appendix B: Genetic algorithms (GAs)

Some well-known meta-heuristics, such as simulated annealing, tabu search, genetic algorithms or neural networks have been proposed to circumvent the local-optimum problem of the

NSMs described above; see also Reeves (1993) for further details. In this paper the Genetic Algorithm (GA) approach will be used to deal with the basic scheduling problem as stated in Sect. 2.

The name *genetic algorithm* originates from the analogy between the representation of a complex structure by means of a vector of components and the genetic structure of a chromosome and its genes.

In selective breeding of plants or animals, offspring are sought which have certain desirable characteristics that are determined at the genetic level by the way the parent's chromosomes combine. In a similar way, in seeking better solutions to complex problems, we often combine pieces of existing solutions. It is expected that following *natural rules*, a set of solutions can be combined so that better ones are obtained.

The major difference with respect to other meta-heuristics is that with GAs a set of solutions, the population, is handled through the optimization process in such a way that along successive generations (iterations) the population fitness will be improved.

GAs were originally developed by Holland (1975) and they proved to be very efficient for solving several combinatorial optimization problems. The natural rules of GAs (see Chelouah & Siarry 2000) are *selection* of the best individuals of a population, *crossover* between two selected individuals to produce two new ones, the sons, which will replace their parents and *mutation*, which arbitrarily changes some characteristics of some individuals. Associated with these rules, three mathematical operators are defined:

Selection operator determines the individuals to be chosen for mating.

Crossover operator determines the manner in which the sons are generated from the parents.

Mutation operator alters some the characteristics of some (very few) individuals from the population.

B.1. Coding the basic scheduling problem

Taking into account that the basic scheduling problem is stated as a permutation problem, the representation of any individual of the population will be its vector of arranged targets σ .

Let n be the population size. This parameter is chosen based on the trade-off between a small value, with brief computational time, and a big value, with a broad coverage of the solution space. Empirical results from many authors suggest that population sizes as small as 30 are adequate, see Reeves (1993).

Another parameter of the GAs is the maximum number of iterations it_M . This parameter can be adjusted from empirical results.

At any iteration $it \in \{0, 1, \dots, it_M\}$, the population is identified by $(\sigma_1^{it}, \sigma_2^{it}, \dots, \sigma_n^{it})$.

At the beginning, when $it = 0$, n random arrangements σ_k^0 , with $k \in \{1, \dots, n\}$, are generated. It is the first population of n individuals.

Each individual σ_k^{it} of the population at iteration it induces (with a greedy procedure) a valid solution σ_k^{it} with N_{sk}^{it} scheduled targets; its associated value is:

$$F_k^{it} = \sum_{i=1}^{N_{sk}^{it}} f(q(S_k^{it}(i))).$$

The overall population at iteration it is also valued with,

$$TF^{it} = \sum_{k=1}^n F_k^{it}.$$

As it is proved by the fundamental *schema theorem*, see Reeves (1993), if it increases the expected total value, $E(TF^{it})$ also increases. Moreover, the optimal solution proposed by the algorithm will be the best-valued individual at any iteration σ^* and this solution is updated through the evolution of the population.

B.2. The GA operators

At iteration $it \in \{0, 1, \dots, it_M - 1\}$, the following operators must be applied:

1. Selection

The quality of the populations is improved each generation it by selecting only those individuals σ_k^{it} , with $k \in \{1, \dots, n\}$ better adapted, i.e. with a higher value of F_k^{it} , the objective function.

A Monte Carlo method is used, the probability p_k of being selected for a given individual σ_k^{it} is defined to be proportional to its relative value with respect to the total value of its generation:

$$p_k = \frac{F_k^{it}}{TF^{it}} \quad \forall k \in \{1, \dots, n\}.$$

With this probability distribution (p_1, \dots, p_n) , n random numbers (i_1, \dots, i_n) , are generated from the set $\{1, \dots, n\}$. Repetitions are allowed. In this way, n individuals of the current population it are selected:

$$\sigma_k^{it+1} = \sigma_{i_k}^{it} \quad \forall k \in \{1, \dots, n\}.$$

After applying this operator, the overall population of n individuals is refreshed with (probably) the best repeated individuals and (also probably) without the worst individuals:

$$(\sigma_1^{it+1}, \sigma_2^{it+1}, \dots, \sigma_n^{it+1}).$$

2. Crossover

Given this refreshed population, the n individuals are classified by $\frac{n}{2}$ couples.

Given a randomly selected couple from the population (σ_k, σ_l) , the two individuals will be crossed and replaced by their offspring with a probability p_c and with probability $(1 - p_c)$ (σ_k, σ_l) will remain unchanged until the next iteration. The crossover operator works in the following way:

- Compute the numbers N_{sk} and N_{sl} of scheduled groups of σ_k and σ_l respectively. Let $N_s = \min\{N_{sk}, N_{sl}\}$.
- Let u be an integer uniform random number in the set $\{1, 2, \dots, N_s\}$.

Table C.1. Main results of the tests of the algorithms (oversubscription factor 1.5).

Input sample ^a	Algorithm ^b	CPU time	F	N_s	$\%N_s(1)$	$\%N_s(2)$	$\%N_s(3)$	$\%N_s(4)$
Exponential-Grey	NSM(Simple)-Random	2.69	12.8	19	47	21	21	11
	NSM(Simple)-Ordered	9.50	57.8	75	47	48	5	0
	NSM(2-opt)-Random	6.32	18.6	27	48	22	22	8
	NSM(2-opt)-Ordered	6.37	57.2	73	48	49	3	0
	NSM(L-K)-Random	10.93	20.1	29	52	17	21	10
	NSM(L-K)-Ordered	12.24	59.9	82	43	44	13	0
	Genetic – C1	1.43	53.1	77	45	35	3	17
	Genetic – C2	1.48	52.7	68	51	41	3	5
Exponential-Standard	NSM(Simple)-Random	13.18	27.5	37	38	59	3	0
	NSM(Simple)-Ordered	4.72	32.5	44	36	61	3	0
	NSM(2-opt)-Random	10.98	30.4	46	35	43	15	7
	NSM(2-opt)-Ordered	5.11	34.9	48	33	65	2	0
	NSM(L-K)-Random	5.22	18.5	28	25	64	7	4
	NSM(L-K)-Ordered	9.17	39.8	57	25	75	0	0
	Genetic – C1	1.48	36.3	66	5	73	22	0
	Genetic – C2	1.43	36.9	73	19	36	27	18
Flat-Grey	NSM(Simple)-Random	0.88	12.2	17	47	29	24	0
	NSM(Simple)-Ordered	0.72	21.0	25	60	40	0	0
	NSM(2-opt)-Random	0.77	15.9	19	63	32	5	0
	NSM(2-opt)-Ordered	1.04	21.3	28	53	29	18	0
	NSM(L-K)-Random	0.60	12.5	19	42	32	10	16
	NSM(L-K)-Ordered	1.71	22.4	33	43	27	30	0
	Genetic – C1	0.55	21.3	31	45	26	26	3
	Genetic – C2	0.87	20.4	25	60	32	8	0
Flat-Standard	NSM(Simple)-Random	0.77	14.4	20	30	70	0	0
	NSM(Simple)-Ordered	0.72	19.3	28	25	71	4	0
	NSM(2-opt)-Random	1.38	15.3	25	20	60	16	4
	NSM(2-opt)-Ordered	0.88	18.7	27	26	70	4	0
	NSM(L-K)-Random	1.65	16.8	25	24	68	8	0
	NSM(L-K)-Ordered	0.66	19.6	28	25	75	0	0
	Genetic – C1	0.66	19.1	31	16	68	16	0
	Genetic – C2	0.66	19.4	31	16	71	13	0

^a Input sample indicates the distribution function of the exposure times in the input list (exponential or flat) and the way in which the quality q has been assigned (either grey or standard policies).

^b Algorithm indicates the type of heuristic used: NSMs (Simple, 2-opt, Lin-Kernigan) or genetic algorithms. It also indicates whether the initial sample has been pre-ordered by right ascension or not for the NSMs algorithms and which of the two crossover operators have been used for the genetic algorithms.

- (c) The first u elements of first (second) offspring are $(\sigma_k(1), \dots, \sigma_k(u)) ((\sigma_l(1), \dots, \sigma_l(u)))$.
- (d) The last $N - u$ elements of first (second) offspring are $(\sigma_l(u + 1), \dots, \sigma_l(N)) ((\sigma_k(u + 1), \dots, \sigma_k(N)))$ avoiding repetitions and maintaining the relative order of $\sigma_k (\sigma_l)$.

For instance, let $N = 9$ be the number of projects approved by the CAT and let σ^k and σ^l be two parents arrangements:

$$\sigma_k = (123 \quad 456789) \quad N_{sk} = 5$$

$$\sigma_l = (739 \quad 146285) \quad N_{sl} = 6.$$

If $u = 3$, then the two offspring are:

$$\sigma'_k = (123 \quad 4685 \quad 79)$$

$$\sigma'_l = (739 \quad 4568 \quad 12).$$

3. Mutation

With probability p_m , each of the individuals of this new population is selected so that two elements $\sigma(k)$ and $\sigma(k')$ of it are permuted. The elements k and k' are integer uniform random numbers in the set $\{1, 2, \dots, N\}$.

In summary, and with these three operators, the family of genetic algorithms is characterized by the parameters vector (n, it_M, p_c, p_m) .

For the scheduling problem analyzed in this article, the genetic algorithm works well with the following parameters:

$$n = 50; it_M = 10; p_c = 0.6; p_m = 0.1;$$

the computations stop when the maximum number of iterations is achieved. Two crossover operator have been used:

1. *Operator C1*: it selects one point X randomly in the $\{1, \dots, N\}$ observations. The first $X - 1$ observations of the

Table C.2. Main results of the tests of the algorithms (oversubscription factor 3).

Input sample ^a	Algorithm ^a	CPU time	<i>F</i>	<i>N_s</i>	% <i>N_s</i> (1)	% <i>N_s</i> (2)	% <i>N_s</i> (3)	% <i>N_s</i> (4)
Exponential-Grey	NSM(Simple)-Random	10.05	17.5	19	84	11	5	0
	NSM(Simple)-Ordered	6.04	38.8	42	81	19	0	0
	NSM(2-opt)-Random	7.03	19.6	22	82	9	5	4
	NSM(2-opt)-Ordered	4.12	38.2	41	83	17	0	0
	NSM(L-K)-Random	12.36	18.9	25	56	24	16	4
	NSM(L-K)-Ordered	4.61	39.8	43	81	19	0	0
	Genetic – C1	1.15	39.4	47	60	40	0	0
	Genetic – C2	1.37	37.8	41	80	20	0	0
Exponential-Standard	NSM(Simple)-Random	9.45	12.8	19	26	63	11	0
	NSM(Simple)-Ordered	4.18	21.7	26	61	35	4	0
	NSM(2-opt)-Random	7.19	15.0	26	31	31	23	15
	NSM(2-opt)-Ordered	7.03	21.7	30	53	17	30	0
	NSM(L-K)-Random	6.42	12.9	20	35	40	15	10
	NSM(L-K)-Ordered	5.60	23.2	30	43	57	0	0
	Genetic – C1	1.21	19.4	38	18	40	2	18
	Genetic – C2	1.32	21.7	43	35	7	28	30
Flat-Grey	NSM(Simple)-Random	1.10	10.6	11	91	9	0	0
	NSM(Simple)-Ordered	0.49	13.6	14	93	7	0	0
	NSM(2-opt)-Random	0.71	7.1	11	37	36	18	9
	NSM(2-opt)-Ordered	0.22	13.0	13	100	0	0	0
	NSM(L-K)-Random	0.60	8.0	10	70	10	10	10
	NSM(L-K)-Ordered	0.22	13.0	13	100	0	0	0
	Genetic – C1	0.50	12.8	15	74	13	13	0
	Genetic – C2	0.49	12.5	14	79	14	7	0
Flat-Standard	NSM(Simple)-Random	0.83	10.2	13	46	54	0	0
	NSM(Simple)-Ordered	0.22	10.6	13	54	46	0	0
	NSM(2-opt)-Random	0.44	6.4	13	8	62	7	23
	NSM(2-opt)-Ordered	0.44	10.3	13	54	38	8	0
	NSM(L-K)-Random	0.87	8.6	15	13	60	27	0
	NSM(L-K)-Ordered	0.93	11.7	20	30	25	45	0
	Genetic – C1	0.49	11.8	15	47	53	0	0
	Genetic – C2	0.49	10.4	14	36	64	0	0

^a As in Table C.1.

first offspring are those of the first parent and the other observations are of the second parent. For this operator, the point *X* has been chosen from the first *N_s* observations to enforce strict changes in the offspring arrangements.

- Operator C2:** it selects two points *X* and *Y* randomly with $X < Y$ in the $\{1, \dots, N\}$ observations, taking the first $X - 1$ observations and the last $N - Y + 1$ observations of the first parent and the other ones of the second parent. This process must be corrected to avoid repeated observations and, anyway, the relative order of the first or second parent is preserved.

Appendix C: Numerical results of the simulations

The main results are summarized in Table C.1 for an oversubscription factor of 1.5 and Table C.2 for an oversubscription factor of 3. The test sample is identified in the first column,

the algorithm and initialization in the second column, the CPU time in the third and the objective function *F* in the fourth. The total number of observations carried out and the percentiles of the observations with $q = 1, 2, 3$ and 4 are given in the following columns. Notice that real oversubscription factors are ~ 2 and ~ 4 due to the time consumed by maneuvering the telescope and setting up the instruments.

References

Ball, M. O., Magnanti, T. L., Monma, C. L., & Nemhauser, G. L. (ed.) 1995, Network Models, Handbooks in Operations Research and Management Science 7 (Amsterdam: Elsevier Science B.V.)
 Chelouah, R., & Siarry, P. 2000, J. Heuristics, 6, 191
 Dasgupta, D., & Michalewicz, Z. (ed.) 1997, Evolutionary Algorithms in Engineering Applications (Springer Verlag, Inc. Berlin)

- Drummond, M., Bresina, J., Swanson, K., Edgington, W., & Henry, G. 1994, The Associate Principal Astronomer Telescope Operatio Model, Proc. in i-SATRAS 94
- Ford, J. 1998, A report on SPIE's International Symposium on Astronomical Telescopes and Instrumentation (URL: <http://www.gb.nrao.edu/GBT/MC/doc/spie98/jford98tripreport/>)
- Garey, M. R., & Johnson, D. S. 1979, Computers and Intractability – A Guide to the Theory of NP-Completeness (New York: W. H. Freeman and Co.)
- Gionnone, G., Chavan, A. M., Silva, D. R., Krueger, A. P., & Miller, G. E. 2000, in Astronomical Data Analysis Software and Systems IX, ed. N. Manset, C. Veillet, & D. Crabtree (San Francisco: ASP), ASP Conf. Ser., 216, 111
- Holland, J. H. 1975, Adaptation in Natural and Artificial Systems (University of Michigan Press, Ann Arbor, MI, Internal Report)
- Johnston, M. D., & Miller, G. E. 1994, in Intelligent Scheduling, ed. M. Zweben, & M. S. Fox (San Francisco: Morgan Kaufmann), 391
- Johnston, M. D., & Adorf, H.-M. 1992, Computers Ops Res., 19 (3/4), 209
- Manset, N., Veillet, C., & Crabtree, D. (ed.) 2000, Astronomical Data Analysis Software and System IX (San Francisco: ASP), ASP Conf. Ser., 216
- Martínez, P., & Yáñez, J. 2002, Neural Networks, 15 (3), 363
- Rayward-Smith, V. J., Osman, I. H., Reeves, C. R., & Smith, G. D. 1996, Modern Heuristic Search Methods (Chichester: John Wiley & Sons Ltd)
- Reeves, C. R. (ed.) 1993, Modern Heuristic Techniques for Combinatorial Problems (Oxford: Blackwell Scientific Publications)