A Survey on the Dynamic Scheduling Problem in Astronomical Observations

Matias Mora[1,2] and Mauricio Solar[1]

[1] Universidad Técnica Federico Santa María,
Computer Systems Research Group, Valparaíso, Chile
`mmora@csrg.inf.utfsm.cl`
[2] Atacama Large Millimeter/submillimeter Array,
Santiago, Chile

Abstract. The tasks execution scheduling is a common problem in computer science. The typical problem, as in industrial or computer processing applications, has some restrictions that are inapplicable for certain cases. For example, all available tasks have to be executed at some point, and ambient factors do not affect the execution order.

In the astronomical observations field, projects are scheduled as observation blocks, and their execution depends on parameters like science goals priority and target visibility, but is also restricted by external factors: atmospheric conditions, equipment failure, etc. A telescope scheduler is mainly in charge of handling projects, commanding the telescope's high level movement to targets, and starting data acquisition. With the growth of observatories' capacities and maintenance costs, it is now mandatory to optimize the observation time allocation. Currently, at professional observatories there is still strong human intervention dependency, with no fully automatic solution so far.

This paper aims to describe the dynamic scheduling problem in astronomical observations, and to provide a survey on existing solutions, opening some new application opportunities for computer science.

1 Introduction: The (Dynamic) Scheduling Problem

In general terms, the scheduling problem is an assignment problem of a number of tasks or jobs, which execution may be restricted by one or more constraints or performance criterion. A survey of the most important results in (multi-criterion) scheduling, and a summary of basic concepts and notations, have been presented in [8], starting with first researches in the mid 50s. The most classical scheduling problem is the *job-shop*, where N ideal jobs are assigned to M identical resources (machines, processors, etc.), while trying to minimize the total execution time.

Besides the static problem, the dynamic scheduling has been frequently researched in recent years. It presents the challenge of processing the schedule on-line, while new jobs are coming into the queue, or new restrictions appear. A complete survey on the dynamic problem for manufacturing systems has been presented in [15], concluding that Multi-Agent Systems are one of the most

M. Bramer (Ed.): IFIP AI 2010, IFIP AICT 331, pp. 111–120, 2010.

promising techniques, although its application needs to be further developed in the future. In general terms, resolution techniques should match a balance between flexibility and robustness.

A simple example of dynamic scheduling can be found in [7]. It is proposed to divide the dynamic job-shop scheduling problem under fuzzy environment into static fuzzy sub-scheduling problems, and uses an improved Giffler & Thompson algorithm is used to solve them. A different approach using a Neuro-Fuzzy Network (NFN) is proposed in [20]. A fuzzy classification of jobs is made in real-time, and then scheduled with a neural network, trained off-line with a genetic algorithm. A simulation shows that this is effective and highly efficient compared to other algorithms (FIFO and Lagrangian Relaxation). As mentioned before, Multi-agent systems have also been used to solve dynamic scheduling decision systems. For example, [3] uses agents to explore more useful decisions in the variable environment, constantly updating the knowledge base. The result is a very flexible algorithm that will adapt itself to the environment. Detected problems are the slow convergence of the algorithm, and the relatively slow response to the environment.

A variation of the dynamic problem is scheduling with dynamic priorities, which has been widely used in CPU and cluster processes scheduling. As described in [11], the main idea is to optimize the processor utilization by dynamic priorities assigned according to deadlines, obligating pending jobs to be executed in a reasonable time. Priorities in these cases are adjusted by the scheduler itself, and not directly by external factors.

Finally, [8] mentions some interesting new development lines, outside the traditional scopes: "interfering job sets" (jobs that have to be run at the same time on the same machine), "scheduling with rejection" (allows the possibility to not run all jobs, by rejecting some of them) and "rescheduling for new orders" (introducing new conditions and jobs). As we will see, all these relatively new developments are related to the problem described in this paper.

2 Astronomical Observations Scheduling

Astronomical observations require specific conditions for their execution, such as the instrument to be used, visibility, sky brightness, etc. All this information, together with the scientific goals of the observation, are presented by an astronomer in a so called *proposal* to apply for observation time. Its format can vary from one institution to another, including the following fields: telescope and instrument (one telescope can work with more than one instrument), main investigator, program description and target(s) list.

In the case of Chile, there are three main institutions managing some of the world's most important telescopes: European Southern Observatory, ESO (La Silla, Paranal, APEX); Association of Universities for Research in Astronomy, AURA (Tololo, Gemini, SOAR); and Observatories of the Carnegie Institute of Washington, OCIW (Las Campanas). Proposals have to be sent to the corresponding Telescope Assignment Committee, which evaluates all proposals, assigning a scientific priority (importance of execution), and approving or rejecting the requested observing times. As most observatories are joint ventures between several organizations and/or countries, the list of approved projects has to comply with the percentage of total time assigned to each part. Normally, telescope time can be applied once per observations period. An observation can be executed in visitor or service mode. Visitor mode observations require the presence of the main investigator (or a collaborator) on site to take the data, while service mode observations are executed by the telescope operator and observatory's science staff members.

Approved proposals will be later scheduled as one or more observation blocks for execution on the corresponding telescope and instrument. The execution of each observation depends on external factors specified in the proposal, and the automatic scheduler and/or telescope operator will have to decide the best time for each case. This problem can consider various thousands of observations per period, although not all targets will be visible during the entire time. Normally, a long term plan will consider factors like visibility over the horizon and moon brightness, while a short term plan will consider specific observation blocks for the next night(s), based on more immediate factors. For a detailed description of a typical observation process at ESO observatories, see [21]. As it could be concluded from this *modus operandi*, it will certainly happen that not all approved observation blocks are executed in a given period. The execution probability of a project will depend on its science priority, but also on the fulfillment of needed conditions for its observation. In order to complete a project, all its observation blocks have to be successfully executed.

The scheduling of astronomical observations is a variation of the dynamic scheduling problem, which has been treated in various ways since several decades by the scientific community. This is a multi-objective problem, as normally various optimizations are required. The most important ones are the maximization of the executed scientific priorities (scientific throughput), and the maximization of observing time usage. This includes minimizing gaps between executions (including readout time, telescope movement to the new source, instrument change and/or calibrations, etc.), and carefully planning required maintenance. Also, the total exposure time of an observation may depend on atmospheric conditions, as it could be necessary to do larger exposures with poor visibility.

Although most of modern professional observatories use certain degree of scheduling automation, there is still a huge part of human intervention to build up the daily plan and to take last minute decisions. External parameters can vary at any time during observations, and therefore a dynamic re-scheduling is needed. If we consider a given execution priority for each block, depending on the quality of observation conditions and importance of scientific goals, external parameters can certainly cause priority changes. Some observations may not be anymore suitable to execute under new conditions. Moreover, as observation blocks depend on target's visibility on the sky, they might be only valid during certain day/night time, and/or certain periods of the year. Therefore, it might happen that initially high priority tasks have to be executed with less priority, or cannot be executed at all within one observations period. Particular observation blocks may also depend on others to be executed. For example, it may be required to execute blocks sequentially or with a certain frequency.

3 Current Approaches

The common astronomy scheduling problem is NP-hard. A good description and basic mathematical model is presented in [2]. In the same publication, the author proposes long and short-term scheduling scopes, and tries to solve the problem using neighborhood search (Lin-Kernighan heuristic) and genetic algorithms. The result is that under certain circumstances (short size and good pre-ordered sample) the neighborhood search can perform better than the genetic algorithm. Nevertheless, the genetic algorithm is in general a better and faster alternative, and does not need a pre-knowledge of the main constraints. The scientific policy (committee) imposes some restrictions that are difficult to handle, depending strongly on the sample characteristics (proposal quality and duration). To take better advantage of automatic scheduling, it is important to have a small degree of over-subscription in the final allocated time, and also a large number of short exposure (duration) project proposals.

Figure 1 shows a typical observations scheduling process, with a list of approved, and *conditionally* approved (over-subscription) observation blocks. The scheduler will sort these blocks according to their execution priority into a queue, and then allocate them in the available time slots. According to varying conditions, this process has to be repeated and the schedule properly adapted.

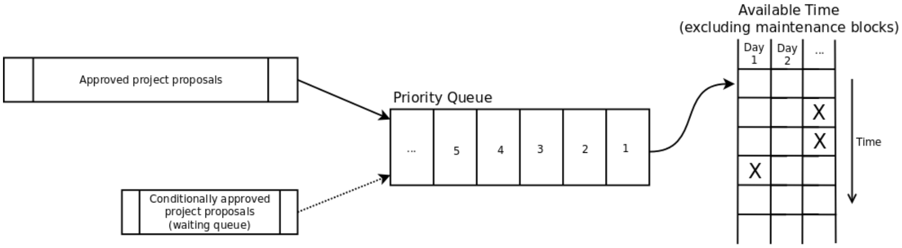

Fig. 1. Example astronomical observations scheduling process *(source: self elaboration)*

The most referenced scheduling solution for astronomical observations is the SPIKE scheduler for the Hubble Space Telescope, developed by the Space Telescope Science Institute (STScI). SPIKE is largely cited as a reference scheduling system, and has also been adapted to other (ground based) telescopes. The current trend is to increase the observations automation, as astronomical projects are getting more complex, observation time more expensive, and decisions more difficult. Some scheduling approaches for current professional projects are discussed below.

3.1 Hubble Space Telescope (HST)

The HST is probably the most famous space telescope, launched in 1990, and best known for its exploration of the deep space from the Earth orbit. It is a collaboration between NASA and the European Space Agency. Space telescopes have the advantage of not depending on atmospheric interference, but are also much more complex to maintain and repair. The HST SPIKE scheduling system, described in [10] and [23], treats the scheduling as a constraint satisfaction problem (CSP), including a toolkit to handle this type of problems. Short and long term scheduling concepts are applied, and several schedule steps are considered: trial assignment heuristic (min-conflicts times), repair heuristic (neural network) and de-conflict (priority selection). Also, rescheduling of observations is possible through the CSP toolkit (task locking and conflict-cause analysis). Since its original implementation in 1987 SPIKE is entirely implemented in Common Lisp and Common Lisp Interface Manager for user interfaces. [13] presents a report about studies related to the generalization of constraint-based scheduling theories and techniques with application to space telescope observation scheduling. For this goal, the Heuristic Scheduling Testbed System (HSTS) planning and scheduling framework was developed. Planning and scheduling are treated as complimentary problems to produce good results. This was validated by producing observation schedules for the HST.

3.2 Very Large Telescope (VLT)

The VLT, operated by ESO, is one of the world's largest optical telescopes, with four 8.2 meter aperture telescopes, designed to work separately or as a single instrument: the VLT Interferometer. It is located in the northern Chilean dessert, on top of the Paranal mountain. The early observations plan for the VLT is discussed in [9], describing the SPIKE scheduling tools, initially developed for the HST, and the requirements to adapt it for VLT use and other ground based observatories. The automated scheduling is thought just as an assistant for human decisions on value judgments, mixing visitor and service mode observations. Nevertheless, it is an important step starting from the visitor only mode in La Silla. Also, first concepts of artificial intelligence are introduced, and envisioned as a good way to significantly reduce calculation times. Later, [21] describes the proposal selection, prioritizing and scheduling process at ESO observatories. Specially interesting is the mention of many factors that normally affect observations, such as time of the year, technical down-time and lunar phase (brightness). Observations are organized in periods of 6 months each.

3.3 Subaru Telescope

This 8.2 meters telescope is located on the top of the Mauna Kea mountain in Hawaii, together with many other large telescopes. [17] describes the (partially implemented) scheduling system for the Subaru Telescope, part of the Subaru Observation Software System (SOSS). It uses SPIKE as scheduling engine, with various extensions to the VLT adaptation, for particular needs. The main objective is to provide "computer-aided planning of observation proposals", and decision support for the Telescope Allocation Committee and the observers.

3.4 Gemini Observatory

The Gemini Observatory is made up of two almost identical 8.1 meters telescopes, located in both hemispheres: Chilean Pachón mountain and Hawaiian Mauna Kea. Simulation results for the Gemini project scheduler are presented in [16]. The general observations plan description is similar to [21], but with less institutional experience. This work focuses on how to distribute observing time in the future, based on ambient factors, but also the partners' proportion. Observable nights are divided into 6-months periods, and A and B types, depending on sky quality.

3.5 Stratospheric Observatory for Infrared Astronomy (SOFIA)

The SOFIA telescope is a collaboration between NASA and the German Aerospace Center. The telescope is mounted on a Boeing 747 aircraft, and observes during flight time. A description of the various computational challenges of the SOFIA project is presented in [6]. In this very special case, scheduling includes flights planning (duration, position and speed, gas consumption, etc.), and also the observations set for each flight. On the previous project of this kind, KAO (Kuiper Airborne Observatory), the entire planning was done by hand and took about 8 hours for each flight. The idea for SOFIA is to assist this process by computational algorithms, rather than replacing human flight planners. In this context, similar experiences from the space are cited: HST, Earth Observing Satellites and Mars Exploring Rovers.

3.6 The Robert C. Byrd Green Bank Telescope (GBT)

The GBT is the world's largest fully movable radio telescope antenna. It is located in West Virginia, and operated by the National Radio Astronomy Observatory (NRAO). A prototype automatic scheduling system has been recently introduced: the GBT Dynamic Scheduling System (DDS), described in [14]. Other than in other dynamic scheduling outlines, this telescope is very interactive and requires the presence of the observing astronomer. Therefore, "the Dynamic Scheduling System is scheduling people rather than scripts", and predictions are an important need, mainly achieved through weather forecasts and weather data from the last four years. The candidates determination for certain periods based on this data is described in [1]. The scoring algorithm in the GBT DDS assigns a priority number to each proposal, taking specially into account the weather predictions, but also stringency and observing efficiency. The actual scheduling algorithms are described by [19], in a 2-phase approach: Sudoku solver for fixed and windowed sessions, and Knapsack algorithm for optimal schedules of remaining time intervals. In conclusion, by dynamic scheduling it is possible to substantially improve telescope time usage efficiency.

4 The ALMA Scheduling Problem

The Atacama Large Millimeter/submillimeter Array (ALMA) is a major collaboration effort between European, North American and East Asian countries, under construction on the Chilean Chajnantor plateau, at 5.000 meters altitude. When completed in 2013 it will be the largest radio-telescope on earth, with more than 60 antennas of 12 and 7 meters diameter, distributed over a wide extension, with up to 16 kilometers of baseline separation. The ALMA interferometer will provide the possibility to be used as a single array, or as up to six (due to limited centralized equipment) minor independent arrays or groups of antennas. As each array is equivalent to one instrument, this can be seen as a multi-telescope problem. Also, the antennas will be changing their position during the year, as different distributions will be used to exploit various kinds of observations. As ALMA is a radio-telescope, observations are not limited to nighttime, and a 24/7 operation with as small downtime as possible is expected.

ALMA will operate exclusively in service mode. Therefore, the Scheduling Subsystem is supposed to provide a fully automatic quasi-real-time dynamic scheduling platform, mostly with the only human participation of supervision. This subsystem is still under development, and the needed algorithm(s) not defined yet. The main problem is the dynamic priorities scheduling, which differs widely from the traditional dynamic job-shop. This particular problem is very similar for all ground based observatories, and ALMA is one real example, that needs this kind of scheduling to accomplish its operations requirements.

A huge part of the telescope operations will be handled through the ALMA software, which is divided into various subsystems, such as Control, Correlator, Pipeline, Archive, etc. (for a complete overview of the ALMA software architecture, see [18]). The Scheduling Subsystem is the one in charge of managing antenna arrays and executing observation blocks dynamically. Its design was first discussed in [22], where a prioritized list of observations is scheduled in real-time. After a critical retrospective of automatic scheduling it is recommended that ALMA should still use human intervention to schedule observations, but assisted by a dynamic scheduler which should be able of adjusting parameters according to weather conditions. A prototype time allocation program called *taco* was implemented and tested. This was not successful, as it wasn't able to adjust parameters (static behavior) and the dynamic decision queue was not suitable for actual observations, as it didn't consider astronomers' availability (which has been later discarded as exclusively service mode will be used). More recently, [5] presented the architecture for the ALMA dynamic scheduler (sections 5 and 7), with emphasis on the scheduler simulation, and some sort of detail about the algorithm implementation (section 6), and current simulated results. This paper is also a good introduction to some basic dynamic scheduling concepts (sections 1 to 4), citing mainly [23]. Furthermore, [12] is an extension to the previous paper, concentrating on scheduling policies and policy factors, like the observations success probability formula.

The Scheduling Subsystem requirements and design are detailed in the "Scheduling Subsystem Design Document" [4]. The current design considers four main parts: Master Scheduler, Project Manager, Scheduler and Simulator (see Figure 2). Observation proposals are represented as projects, divided into Scheduling Blocks (SB), which are handled as a queue by the Master Scheduler. The queue is created and maintained by the Project Manager, which will also start the reduction pipeline once an observation is completed. SBs from the master queue are assigned to different Schedulers, which will execute them on an associated antennas array (one of these Schedulers is assigned to one array). A Scheduler will be able to operate in interactive or dynamic (automatic) mode. Additionally, the Simulator will be used to simulate the overall scheduling behavior, for planning purposes and performance tuning. The dynamic mode Schedulers are the main problem, which goal consist in providing a full automation of SB execution, with manual override capabilities. The general requirements of the ALMA scheduler include to be able to handle about 18.000 SBs per period ($\sim$ 12 months), and to do a rescheduling process in much less than 1/2 of a typical scheduling block length ($\sim$ 30 minutes). The Scheduling Subsystem is currently under development, and the defined architecture could still change in the near future.

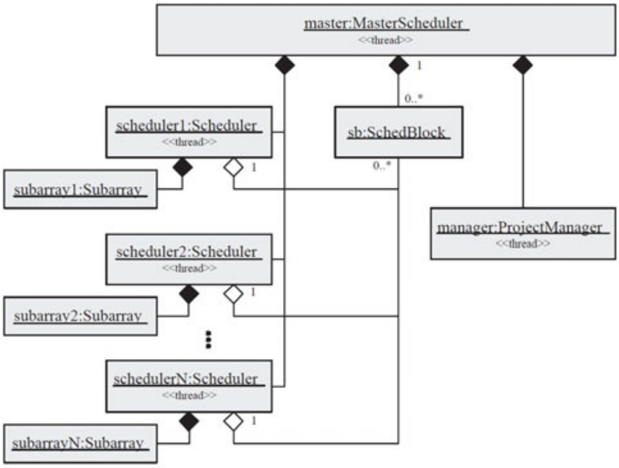

Fig. 2. ALMA scheduling system *(source: [5])*

As each array can be formed by any combination of antennas (different combinations make actually a difference depending on the physical positions), the number of possible junctions for a 1-array problem is in theory $O(n!)$, with n the number of available antennas. Nevertheless, it is expected that most of the major observations will use all available antennas to exploit their full potential. A simplification of this kind would be still a very useful approach.

Different alternatives need to be analyzed for the resolution of the ALMA problem, first considering the automation requirement, but also the distributed nature of the project, mainly regarding the varying antenna locations. Although this problem differs from the manufacturing job-shop, we can see some correlation in the recent development of both areas, in terms of changing environment and even the distribution of jobs and processing machines. It would be therefore recommendable to start with techniques that have been successful for distributed job-shops, like Multi-Agent Systems (see [15]).

5 Conclusions

The astronomical observations scheduling problem has been presented, together with some of its most extreme variants, and a current example case description for the ALMA project. As it has been identified, this new problem considers some non-traditional elements, like interfering jobs, rejection possibility and rescheduling for new order.

None of the existing professional astronomical scheduling solutions provides a full automatic scheduler under changing conditions and for several telescopes, and none of the existing algorithms has solved the dynamic priorities problem for this case. As new astronomical projects will depend on scheduling optimization to make the best use of limited and expensive observing time, this problem is an interesting research topic to be analyzed from many different views. The new conditions for the scheduling problem are the major engineering challenges for this case.

This is just a first overview, to motivate further work on this topic. A first approach of specialization in this area, for current and future projects, is being done by the Computer Systems Research Group (CSRG) at Universidad Técnica Federico Santa María in Chile, in collaboration with the ALMA Scheduling Subsystem development team and the UTFSM Informatics Department.

Acknowledgements

The authors would like to thank the revisions and contributions by UTFSM and ALMA colleagues, and the ALMA-Conicyt Fund #31080031.

References

[1] Balser, S., et al.: GBT Dynamic Scheduling System: Algorithms, Metrics and Simulations. In: Astronomical Society of the Pacific Conference Series. Astronomical Data Analysis Software and Systems XVIII, vol. 411, p. 330 (2009)

[2] de Castro, A.I.G., Yanez, J.: Optimization of telescope scheduling: Algorithmic research and scientific policy. Astronomy and Astrophysics 403, 357–367 (2003)

[3] Chao, W., Zhen-qiang, B., Chang-yi, L., Wen-yu, B.: A Self-adaptive Knowledge Based Dynamic Scheduling Decision Mechanism. In: International Conference on Management Science and Engineering, pp. 463–467 (October 2006)

[4] Clarke, D., Lucero, S., Farris, A.: Scheduling Subsystem Design Document. Tech. Rep. COMP-70.55.00.00-003-E-DSN, National Radio Astronomy Observatory, Socorro, NM (May 2009)

[5] Farris, A.: Scheduling simulation in ALMA. In: Proceedings of SPIE 2004. Optimizing Scientific Return for Astronomy through Information Technologies, vol. 5493, pp. 302–312 (June 2004)
[6] Frank, J.: SOFIA's challenge: automated scheduling of airborne astronomy observations. In: Second IEEE International Conference on Space Mission Challenges for Information Technology, SMC-IT (2006)
[7] Fu-ming, L., Yun-long, Z.: Study on the Job Shop Dynamic Scheduling under Fuzzy Environments. In: International Conference on Computational Intelligence and Security, vol. 1, pp. 90–96 (November 2006)
[8] Hoogeveen, H.: Multicriteria scheduling. European Journal of Operational Research 167, 592–623 (2005)
[9] Johnston, M.D.: Automated observation scheduling for the VLT. Tech. rep., Space Telescope Science Inst., Baltimore (1988)
[10] Johnston, M.D.: Spike: AI scheduling for NASA's Hubble Space Telescope. In: Sixth Conference on Artificial Intelligence Applications, vol. 1, pp. 184–190 (1990)
[11] Liu, C.L., Layland, J.W.: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. Journal of the ACM 20, 46–61 (1973)
[12] Lucero, S., Farris, A.: Dynamic Scheduling in ALMA. In: Astronomical Society of the Pacific Conference Series. Astronomical Data Analysis Software and Systems XVI, vol. 376, p. 673 (2007)
[13] Muscettola, N., et al.: Investigations into Generalization of Constraint-Based Scheduling Theories with Applications to Space Telescope Observation Scheduling. Tech. rep., The Robotics Inst., Carnegie-Mellon University, Pittsburgh (1996)
[14] O'Neil, K., et al.: The GBT Dynamic Scheduling System: A New Scheduling Paradigm. In: Astronomical Society of the Pacific Conference Series. Astronomical Data Analysis Software and Systems XVIII, vol. 411, p. 147 (2009)
[15] Ouelhadj, D., Petrovic, S.: A survey of dynamic scheduling in manufacturing systems. Journal of Scheduling 12, 417–431 (2009)
[16] Puxley, P.J.: Execution of queue-scheduled observations with the Gemini 8-m telescopes. In: Proceedings of SPIE 1997. Telescope Control Systems II, vol. 3112, pp. 234–245 (July 1997)
[17] Sasaki, T., et al.: Observation scheduling scheme for the Subaru telescope. In: Proceedings of SPIE 2003. Advanced Telescope and Instrumentation Control Software, vol. 4009, pp. 350–354 (June 2003)
[18] Schwarz, J., Farris, A., Sommer, H.: The ALMA software architecture. In: Proceedings of SPIE 2004. Advanced Software, Control, and Communication Systems for Astronomy, vol. 5496, pp. 190–204 (June 2004)
[19] Sessoms, E., et al.: The GBT Dynamic Scheduling System: Scheduling Applications of the Knapsack Problem and Sudoku. In: Astronomical Society of the Pacific Conference Series. Astronomical Data Analysis Software and Systems XVIII, vol. 411, p. 351 (2009)
[20] Shugang, L., et al.: Job shop scheduling in real-time cases. The International Journal of Advanced Manufacturing Technology 26, 870–875 (2005)
[21] Silva, D.R.: Service mode scheduling at the ESO Very Large Telescope. In: Proceedings of SPIE 2002. Observatory Operations to Optimize Scientific Return III, vol. 4844, pp. 94–103 (June 2002)
[22] Wright, M.C.H.: Dynamic Scheduling. Tech. rep., Radio Astronomy Laboratory, University of California, Berkeley (December (1999)
[23] Zweben, M., Fox, M.S.: SPIKE: Intelligent Scheduling of HST Observations, ch. 14, pp. 391–422. Morgan Kaufmann Publishers, San Francisco (1994)