SPACE
TELESCOPE
SCIENCE
INSTITUTE

PREPRINT SERIES
No.307

AUTOMATED OBSERVATION SCHEDULING FOR THE VLT

Mark D. Johnston

August 1988

(NASA-CR-[illegible]) AUTOMATED OBSERVATION SCHEDULING FOR THE VLT (Space Telescope Science Inst.) [illegible] CSCL 03A
N89-1[illegible]
Unclas
G3/89 [illegible]

SPACE TELESCOPE SCIENCE INSTITUTE
3700 San Martin Drive Baltimore, MD 21218

Operated by the Association of Universities for Research in Astronomy, Inc., for the National Aeronautics and Space Administration

AUTOMATED OBSERVATION SCHEDULING FOR THE VLT

Mark D. Johnston

Space Telescope Science Institute
3700 San Martin Drive
Baltimore, MD 21218

Published in the proceedings of the

Conference on Very Large Telescopes and Their Instrumentation

Garching, March 1988

European Southern Observatory

AUTOMATED OBSERVATION SCHEDULING FOR THE VLT

Mark D. Johnston
Space Telescope Science Institute
3700 San Martin Drive
Baltimore, MD 21218 USA

Abstract: It is becoming increasingly evident that, in order to optimize the observing efficiency of large telescopes, some changes will be required in the way observations are planned and executed. Not all observing programs require the presence of the astronomer at the telescope: for those programs which permit "service observing" it is possible to better match planned observations to conditions at the telescope. This concept of "flexible scheduling" has been proposed for the VLT: based on current and predicted environmental and instrumental conditions, a flexible scheduler would help the telescope operations staff select and execute observations which make the most efficient possible use of valuable telescope time. A similar kind of observation scheduling is already necessary for some space observatories, such as Hubble Space Telescope (HST). Space Telescope Science Institute is presently developing scheduling tools for HST, based on the use of "artificial intelligence" software development techniques. These tools could be readily adapted for ground-based telescope scheduling since they address many of the same issues. This paper describes the concepts on which the HST tools are based, their implementation, and what would be required to adapt them for use with the VLT and other ground-based observatories.

> "All too frequently at the moment someone travels to La Silla for a programme which requires excellent seeing to have only some mediocre nights, while another astronomer a week later experiences superior conditions which his programme does not need. With the VLT such wasteful procedures cannot be accepted."
>
> -- *Proposal for the Construction of the 16m Very Large Telescope*

1 Introduction

Optimizing telescope utilization is an important but difficult problem. Observing time on large telescopes is a scarce resource: oversubscription by factors of several are already typical, and this is hardly likely to decrease with the construction of newer and more advanced facilities. It is thus important to consider how the utilization of existing and planned telescopes can be increased to the maximum extent possible.

The simplest mode of telescope operation is the "classical" one: allocate fixed blocks of time to individual programs to be carried out by the astronomer who travels to the telescope for this purpose. This mode has the advantages that advance planning is possible, and that observers

---

[1] To appear in *Proceedings of the ESO Conference on Very Large Telescopes and their Instrumentation*, Garching, March 1988.

are in control of their own observations and, except for weather, are primarily responsible for their quality. The drawbacks of this mode are well known: too much is left to the vagaries of the weather, and both individual programs and overall observatory efficiency can suffer as a result. The most obvious way to improve this situation is to move towards an integrated mode of operation in which nights are not pre-allocated to specific programs but can be scheduled dynamically as conditions warrant ("flexible scheduling"). In this way observations can be matched to the prevailing environmental and instrumental conditions. Not only does this promise to increase the effective utilization of the telescope, but it also increases the chances that any individual program will be carried out under its most appropriate conditions. This latter point is especially important for programs with the most stringent observing requirements. Some programs will continue to require the presence of the observer at the telescope, so that in practice some mixture of classical and integrated operation is likely to evolve. Remote observing offers an important way to bridge these two modes, especially if remote observing stations become sufficiently widely available so that travel and advance planning for access to them can be minimized.

So long as the responsibility for scheduling and executing observations is vested solely in the individual astronomer, observatory-level telescope scheduling can be limited to the relatively simple problem of allocating blocks of time to each accepted program. The development of flexible scheduling will greatly expand the scheduling needs of an observatory, to the point that it is not reasonable to expect operations staff to schedule telescope usage without software support. It is therefore important to consider what elements are necessary to provide this support, and how automated scheduling can become an effective part of overall telescope operation.

2 Aspects of Telescope Scheduling

There are several aspects of telescope operations that are related to scheduling, ranging from the initial allocation of observing time to the scheduling decisions made within one night by the observer. What is common to these various aspects is the existence of a pool of candidate activities (observations, programs, instrument tests, etc.) to be scheduled over some time period. What distinguishes one aspect from another are the goals and constraints that act as dominant scheduling factors.

From an operations point of view, telescope scheduling can be roughly divided into long-term and short-term problems. The long-term problem is concerned with construction of an overall scientific program for one or more observing periods, without necessarily considering the details of which observations are scheduled on which night. This process is not often thought of as scheduling, but in fact it is essentially the construction of a high-level integrated schedule incorporating the scientifically highest-ranked proposals, balanced against available resources and satisfying institutional or policy objectives. In this context the activities to schedule will generally be whole programs, although blocks of reserved time or individual observations could equally well be considered.

Short-term scheduling can be viewed as the process of deciding on a sequence of individual observations to schedule over a more limited time range. In general a short-term schedule implements in detail a portion of some long-term schedule, on which it is based and to which it must conform. The nature of the short-term scheduling problem depends greatly on the "mode" of observation, where by this we distinguish "classical" observing (carried out by the astronomer who travels to the telescope for a run of a specified number of nights) from the more recently developed "service" or "absentee" observing (carried out by on-site telescope operations staff from specifications provided by the proposing astronomer) and from "remote" observing (carried out the by the astronomer, but via remote control from a site more convenient than the telescope itself). Experiments with service and remote observing have been conducted at several observatories (e.g. [1,2]) and it is expected that these modes will become more and more widely used. Time for these observing modes will thus have to be allocated for each observing period, along with the nights for classical observing.

Long-term and short-term scheduling are of course intimately related. Both are subject to the same basic telescope and operational constraints. Both schedule the same activities but at different levels of granularity. It is important to devise long-term schedules which can be implemented when considered in detail, just as it is important that short-term schedules satisfy the overall boundary conditions imposed by a long-term schedule. These differences in timescale and level of granularity are not fundamental: as discussed further below, long-term and short-term scheduling can be regarded as different views of essentially the same process and could well make use of the same underlying scheduling software.

The short-term scheduling "horizon" is fundamentally limited by the unpredictability of the weather: there is therefore no point in constructing in advance short-term schedules for extended periods of time. This is the motivation for flexible scheduling, i.e. short-term scheduling (and re-scheduling) conducted on the same timescale as changes in the weather. The ability to deal with the unpredictability of the telescope environment, and to effectively handle schedule disruptions when they inevitably occur, are essential components of any telescope scheduling system.

It is evident that the greatest efficiency gains from automated scheduling will be obtained when the greatest flexibility exists to exploit good scheduling opportunities as they arise. There are three main requirements on this flexibility:

1. the physical capability to respond to schedule changes (e.g. rapid instrument changeovers). This factor is well recognized and, e.g., is being designed into the VLT from the outset [3]. It will not be considered further here.
2. a sufficiently rich pool of candidate observations that a probable good match can be found to current environmental conditions.
3. the absence of prior commitments that would forbid taking advantage of new and better scheduling opportunities when they occur.

Clearly, criteria (2) and (3) above are most satisfactorily met in service observing mode, where a relatively large pool of observations is available to be scheduled at the discretion of the telescope operations staff. Classical observing offers the fewest opportunities for schedule optimization because of the generally limited number of choices available to respond to varying observing conditions. To the extent that remote observing is simply classical observing via remote control, schedule optimization is also very limited. However, since the travel requirements to a remote observing station are not as strict as to the telescope site itself, it can be expected that some degree of flexibility could be incorporated into remote observing which would be impractical in the pure classical observing mode.

3 Automated Scheduling

In this section we first discuss the basic elements and capabilities that are required of a telescope scheduling system, and then consider various approaches to implementing these elements.

3.1 Scheduling Tools: Concept

The approach we take here is that automated scheduling is fundamentally a support tool for the people who are responsible for making scheduling decisions. Automated scheduling software can provide this support by rapidly evaluating scheduling choices based on appropriate constraint and preference criteria. It is essential to realize from the start that these criteria basically represent human decision rules and value judgements.

In this approach one of the most important characteristics of a scheduling system is how it interacts with the user. The user must have visibility into all aspects of the scheduling problem and the evolving schedule. The user must also have control, i.e. the ability to override any decisions made by the scheduling software, and the ability to create and evaluate alternative schedule fragments. Because of the large volume of information required to specify even modest-sized realistic scheduling problems, it is almost essential to utilize graphical display and interaction capabilities. This leads to the concept of implementing scheduling tools on single-user workstations, where high-speed graphics and dedicated processing power can both be exploited.

Fundamental to the operation of any scheduling system is an adequate description of precisely what activities are to be scheduled. Not only the activities but also their statuses and the relationships among them must be known. This implies the existence of what is essentially an up-to-date database of activities to schedule.

A realistic scheduling system must have the ability to handle a rich variety of scheduling constraints, where by constraint we mean here any factor that affects when an activity can or should be scheduled. It is not very difficult to focus on only one or a few aspects of schedule optimization and find adequate solutions, only to discover that essential factors have been left out of consideration and do not fit into the chosen scheduling framework. Constraints can be roughly classified into three major categories:

- absolute constraints depend only on time and not on when other activities are scheduled. Examples of this type include target visibility constraints, moon brightness, and eclipse times in binary systems.
- relative constraints represent explicit dependencies of an activity on when one or more other activities are scheduled. Examples of this type include precedence (order) constraints and minimum and maximum time separation constraints.
- resource constraints specify implicit mutual dependencies of activities on each other. Included in this category are resource availability and capacity constraints.

Constraints in all three categories may be either "strict" or "preference": strict constraints cannot be violated under any circumstances; preference constraints specify conditions that are more desirable than others to appear in the schedule. Degrees of preference are of course common and must be represented.

Since the primary purpose of automating telescope scheduling is to optimize telescope utilization, it is clearly important that a scheduling system adequately represent what is meant by "optimal". This is less straightforward than it might seem at first: scheduling goals vary depending on the circumstances, so that a schedule which is optimal in some sense can be far from optimal in another. For example, at different times the most important optimization criterion could be some combination of overall telescope throughput, picking up a disrupted schedule, diagnosing an instrument problem, and scheduling a best match to changing environmental conditions. It is thus important that a scheduling system be flexible in terms of the high-level criteria by which schedule optimality is judged.

The capability to weigh and balance conflicting constraints and goals is implicit in the discussion of constraints and optimization above, but is worth noting separately as a major area that must be addressed. Strict constraints can be exploited to help reduce the search required to find optimal schedules: schedules that violate strict constraints can be quickly eliminated from consideration. This in itself, however, does nothing to solve the problem of conflicting preference constraints. For example, some balance must be struck between high priority observations and those which better match current seeing conditions. Resolution of conflicting preferences is one of the core issues that must be addressed by any scheduler.

On the more practical side, system flexibility and throughput are both important considerations. Scheduling problems are not static. While many of the most important scheduling constraints and goals can be well specified and will not change, others will arise as a result of operational experience with different types of observations. It should therefore be straightforward to "teach" the scheduling system about new constraints, goals, and optimization criteria without a major redesign effort. Throughput is especially important in telescope scheduling where the unpredictability of the weather will demand frequent reactive scheduling as well as the ability to maintain a "grid" of simultaneous alternative schedules for different weather conditions.

Finally, we note that none of the above criteria make any essential distinction between long-term and short-term scheduling: these differ only in that different types of activities, constraints, and optimization criteria are relevant.

3.2 Approaches to Scheduling Software

Computer techniques for optimal scheduling have been investigated for a number of applications (see, e.g., [4] for a comprehensive review and bibliography). Much of this classical work has focused on versions of the idealized "job-shop" scheduling problem, i.e. the problem of scheduling n tasks on m machines. This problem and related ones are NP-complete, meaning essentially that there are no efficient algorithms for finding optimal solutions (see, e.g. [5]). Much of the work on finding approximate solutions cannot be readily applied to "real" scheduling problems (including telescope scheduling) because of the large number of simplifying assumptions that must be made and because only very simplistic optimization criteria are permitted.

In recent years a variety of new software methodologies have been developed under the general term of "artificial intelligence" (AI). This refers to a collection of software development techniques and tools that have evolved in the course of computer science research as effective ways to represent and solve certain kinds of problems. These techniques have moved from the laboratory into widespread use in applications as their effectiveness has been demonstrated. For the purposes of automated scheduling, the most important of these are: a language (Lisp) that is particularly appropriate for manipulating complex data structures and symbolic data; object oriented programming with inheritance and message passing; rule-based programming facilities; integrated graphics and window tools; and a rich software development environment.

Several artificial intelligence research efforts have considered scheduling as a domain where AI techniques can be fruitfully applied. Of particular interest is the factory scheduling work of Fox, Smith, and co-workers [6,7,8] who have developed a rich constraint representation and versatile reasoning process for attacking realistic factory scheduling problems. While factory scheduling shares a number of common features with telescope scheduling (most notably a similar set of precedence and efficiency constraints), there are some important differences. The most significant of these is the enormous variation in the degree of predictability of critical scheduling constraints, e.g. from the weather at one extreme to the motions of celestial objects at the other.

At the Space Telescope Science Institute (STScI) we have for some time been working on a project (SPIKE) to apply AI software technology specifically to the problem of scheduling Hubble Space Telescope [9,10]. HST scheduling is an extremely demanding task [11,12,13], requiring the scheduling of some tens of thousands of exposures per year. These exposures are subject to a large number of scheduling constraints [14,15], some derived from the scientific goals of the proposer, some a consequence of HST design, operating characteristics, and low earth orbit environment. Because HST operates almost entirely in a pre-planned mode, detailed short-term schedules must be defined weeks ahead of time. These schedules are integrated, in that exposures from many different proposals may be scheduled to occur during a single day of observing.

As part of the SPIKE project we have developed a framework for representing and reasoning with the multiplicity of constraints that enters into astronomical observation scheduling [16]. Associated with each activity to be scheduled (typically an exposure or collection of exposures) is a "suitability function", a function of time whose value represents how desirable it is to start an activity at that time. Suitability functions are derived from constraints, an arbitrary number of which may be associated with each activity depending on the type of activity and any specific factors that can affect when it is scheduled. The suitability function of an activity is the product of all of the suitability functions derived from its constraints. This not only mirrors an intuitive notion of how to combine different sources of evidence for and against for scheduling an activity at a given time, it can also be shown to be logically required by the plausible assumptions that the combination of evidence should be associative and monotonic [17,18,16].

Suitability functions provide for "low-level" reasoning about scheduling constraints and preferences: there remains the problem of searching the enormous space of scheduling possibilities to find optimal scheduling choices. This process, referred to as "strategic" scheduling, has been approached in three ways in SPIKE [13]:

- procedural search: this includes standard search techniques such as best-first or most-constrained-first algorithms. These approaches tend to be computationally expensive and often encounter dead-ends which result in grossly sub-optimal schedules.

- rule-based scheduling: one way to represent high-level strategic scheduling knowledge is in the form of rules. For example, it is possible to write almost verbatim a rule of the form "if there is an unscheduled high priority activity which is highly constrained and related to activities already scheduled, then try to schedule it next." This form of control allows for easy incorporation of new scheduling strategies as experience is gained with complex scheduling situations.
- neural networks: a very different approach makes use of an "artificial neural network" [19,20,21] to represent a set of discrete scheduling choices [22,23]. These networks are conceptually composed of a large number of simple processing elements operating in parallel whose computational power comes from their massive interconnection. These connections can be derived directly from the suitability functions of the activities to schedule. The advantages of this approach are rapid execution, the ability to easily reschedule, and, on hardware now in the development stage, the possibility for a true parallel implementation.

On the basis of our experience in developing SPIKE we have concluded that AI software development methodology provides an extremely powerful means with which to attack scheduling problems. The advantages of using these techniques are primarily a rapid software development cycle, a concise but expressive representation of scheduling data, flexibility in the definition and modification of scheduling constraints, and the ability to incorporate a graphics-oriented user interface to help the scheduler understand and modify the schedule.

3.3 Ground-based Telescope Scheduling with SPIKE: An Experiment

The development of the SPIKE scheduling system has followed closely the general principles described in Section 3.1, but, for obvious reasons, has focused closely on the specific constraints most relevant to the HST scheduling problem. Since one of these general principles is flexibility, we have conducted a experiment designed to test this aspect of our approach by applying SPIKE to a problem very different from that of HST scheduling, namely the problem of scheduling the ESO 3.6m telescope in Chile [24]. For the trial period (41), this problem consists of 50 ("classical" mode) programs to be scheduled in 183 nights. Each program is subject to strict and preference constraints on month (first and second priority), and on dark, gray, or bright time. A few programs also have additional absolute timing constraints or participate in relative constraints on order and time separation. A constraint was also included that the "cost" of switching between an optical and IR instrument was a night of setup and calibration time.

Figure 1 shows a copy of the workstation screen illustrating some of the programs being scheduled in this experiment. The central window (which is partly obscured) shows the values over a 6-month time span of the suitability functions for a sample of programs. These represent the preferences for month and moon phase. The bottom window is another view of the same information but at a higher time resolution. The top window is a snapshot of the neural network in operation: each grid point (neuron) represents the scheduling of a program (vertical axis) to start on a specific night (horizontal axis). In this example not all programs have been scheduled.

What is interesting to note is that none of the existing SPIKE scheduling software had to be modified in order to handle this problem: three new constraints were defined (month preference, moon phase, and optical/IR switching time) along with one new type of activity (program).

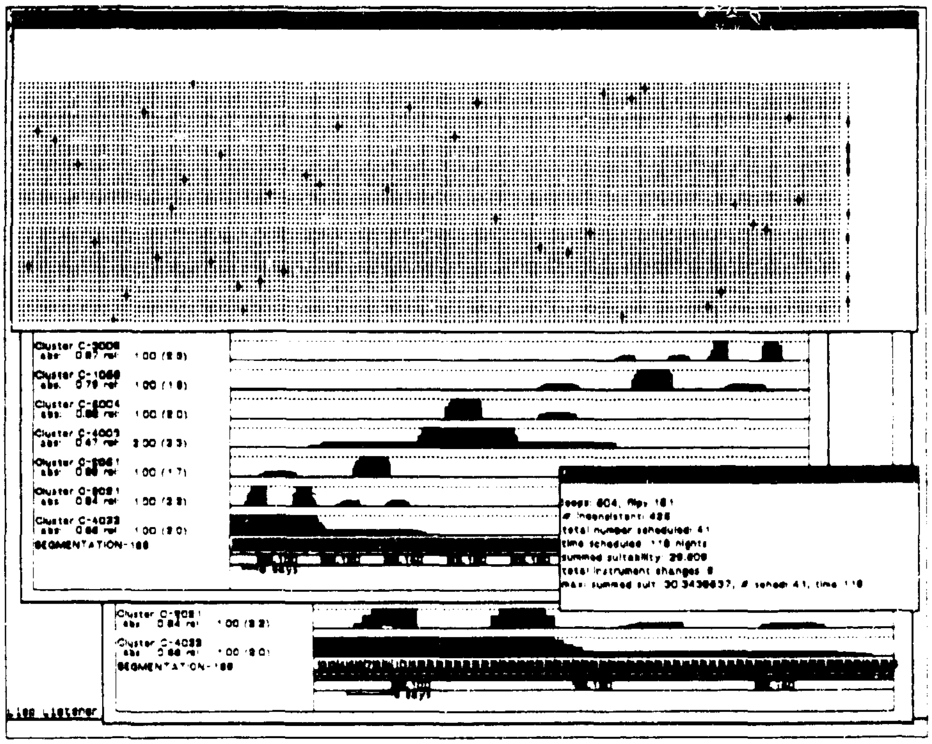

Figure 1: Workstation screen showing the HST scheduling tools at work on programs for the ESO 3.6m telescope in La Silla.

While this experiment is still in progress, particularly on strategic approaches to generating complete schedules, the results so far are very encouraging.

4 Discussion

It is clear that software technology and approaches to scheduling have reached a sufficient level of development that automated telescope scheduling is a realistic goal. The use of artificial intelligence techniques makes it possible to develop and adapt software, such as the HST SPIKE scheduling tools, for a variety of telescope scheduling problems. For the next generation of astronomical observatories, now in the design stage, automated scheduling offers a significant potential for increases in observing efficiency and telescope utilization. There remain, however, a number of issues that must be addressed before automated scheduling can be successfully integrated into the routine operation of large telescopes:

- A flexible scheduling system must be aware not only of what remains to be scheduled but also of what has happened and of current and predicted environmental conditions. This means that the scheduler must be integrated into the overall operational environment in such a way that this information is readily accessible (see, e.g. [25])

- Optimal scheduling requires the existence of a pool of observations not all of which can be executed (see, e.g., [26]). It must be accepted by the community that, in order to optimize the overall observing program, some individual programs may not be scheduled. STScI has allowed for this by accepting proposals at a priority level of "supplemental", but so far no one has been disappointed by being accepted at this priority level ar [illegible] ing their program fail to be executed.
- For scheduling software to exploit preferences and constraints to generate [illegible] schedules it must know about them, which means that they must be specified by [illegible] or be derivable from information implicit in the proposal. Significantly more detail may be required in future proposals than has been necessary in the past. STScI has simplified this process by providing a remote proposal submission system which accepts machine-readable observing proposals submitted over a computer network[27]. Facilities of this type are likely to be necessary for groundbased telescopes which plan to make extensive use of automated scheduling capabilities.

Acknowledgements: The author is grateful to G. Miller, J. Sponsler, and S. Vick (STScI), D. Rosenthal (NASA Ames Research Center) and H.-M. Adorf (ST-ECF) for many useful discussions, and for the hospitality and support of the Space Telescope European Coordinating Facility and the European Southern Observatory (Garching) where some of this work was conducted. Special thanks are due to J. Breysacher (ESO) for useful discussion on the ESO scheduling process and for making available ESO proposal data.

Space Telescope Science Institute is operated by the Association of Universities for Research in Astronomy for the National Aeronautics and Space Administration.

References

[1] Longair, M.S., Stewart, J.M., and Williams, P.M. 1986: "The UK Remote and Service Observing Programmes," Quarterly Journal of the Royal Astronomical Society **27**, p. 153

[2] Raffi, G. 1988: "Remote Observing," in *Proc. ESO Conference on Very Large Telescopes and their Instrumentation*, Garching, March 1988, this volume.

[3] *Proposal for the Construction of the 16m Very Large Telescope*, ESO, Garching, 1987.

[4] King, J.R., and Spachis, A.S. 1980: "Scheduling: Bibliography and Review," International Journal of Physical Distribution and Materials Management **10**, p. 105

[5] Garey, M., and Johnson, D. 1979: *Computers and Intractability*, (W.H. Freeman & Co: San Francisco).

[6] Fox, M. 1983: "Constraint-Directed Search: A Case Study of Job Shop Scheduling," Ph D. Dissertation, Computer Science Dept., Carnegie Mellon University

[7] Fox, M, and Smith, S. 1984: "ISIS: A Knowledge-Based System for Factory Scheduling," Expert Systems **1**, p. 25.

[8] Smith, S., Fox, M., and Ow, P 1986: "Constructing and Maintaining Detailed Construction Plans," AI Magazine. Fall 1986, p 45

ORIGINAL PAGE IS
OF POOR QUALITY

[9] *The Space Telescope Observatory*, 1982, NASA CP-2244, ed. D. Hall

[10] *Call for Proposals*, October 1985, Space Telescope Science Institute

[11] Miller, G., Rosenthal, D., Cohen, W., and Johnston, M. 1987: "Expert System Tools for Hubble Space Telescope Observation Scheduling," in *Proc. 1987 Goddard Conference on Space Applications of Artificial Intelligence*; reprinted in Telematics and Infomatics **4**, p. 301 (1987).

[12] Johnston, M., 1988: "Automated Telescope Scheduling," in *Proc. Conf. on Coordination of Observational Projects*, Strasbourg, Nov. 1987, in press.

[13] Miller, G., Johnston, M., Vick, S., Sponsler, J., and Lindenmayer, K. 1988: "Knowledge Based Tools for Hubble Space Telescope Planning and Scheduling: Constraints and Strategies", in *Proc. 1988 Goddard Conference on Space Applications of Artificial Intelligence*.

[14] Sherrill, T.J. 1982: *Space Telescope Design Reference Mission*, Lockheed Missiles and Space Company, Inc.

[15] "HST Planning Constraints", 1987, Space Telescope Science Institute, SPIKE Report 87-1.

[16] Johnston, M. 1988: "Reasoning with Scheduling Constraints and Preferences," in preparation.

[17] Good, I. 1985, "Weight of Evidence: A Brief Survey" in *Bayesian Statistics 2*, Ed. Bernardo, J., DeGroot, M., Lindley, D., and Smith, A. (North-Holland: Amsterdam), p. 249.

[18] Spiegelhalter, D. 1986: "A Statistical View of Uncertainty in Expert Systems" in *Artificial Intelligence and Statistics*, ed. Gale, W. (Addison-Wesley), p. 17.

[19] Hopfield, J. 1982: "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Natl. Acad. Sci. USA **79**, p. 2254.

[20] Hopfield, J., and Tank, D. 1985: "Neural Computation of Decisions in Optimization Problems," Biological Cybernetics **52**, p. 141.

[21] Jeffrey, W., and Rossner, R. 1987: "Optimization Algorithms: Simulated Annealing and Neural Network Processing," Ap. J. **310**, p. 473

[22] Adorf, H.-M., and Johnston, M. 1988: "Artificial Neural Nets in Astronomy", in *Proc Conf. "Workshop Konnektionismus"*, ed. C. Lischka and J. Kindermann, Bonn, Feb. 1988, to appear.

[23] Johnston, M., and Adorf, H.-M. 1988: "Scheduling with Neural Networks", in preparation

[24] Breysacher, J. 1988: "The Observing Time Distribution in Major Ground based Observatories: A Complex Task," in *Proc. Conf. on Coordination of Observational Projects*, Strasbourg, Nov. 1987, in press

[25] Fosbury, R.A.E., Adorf, H.-M., and Johnston, M. 1988: "VLT Operations - the Astronomers' Environment," in *Proc. ESO Conference on Very Large Telescopes and their Instrumentation*, Garching, March 1988, this volume.

[26] Van der Laan, H. 1984: quoted in *Very Large Telescopes, their Instrumentation and Programs*, IAU Colloq. No. 79, April 1984, ESO, Garching

[27] Jackson, R., Johnston, M., Miller, G., Lindenmayer, K., Monger, P., Vick, S., Lerner, R., and Richon, J. 1988: "The Proposal Entry Processor: Telescience Applications for Hubble Space Telescope Science Operations", in *Proc. 1988 Goddard Conference on Space Applications of Artificial Intelligence*.