

# Queue Scheduling the Alan Cousins Telescope

by

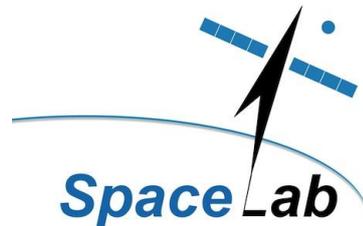
*Deneys Sean Maartens*



SUBMITTED TO THE  
FACULTY OF ENGINEERING & THE BUILT ENVIRONMENT  
UNIVERSITY OF CAPE TOWN

in partial fulfilment of the requirements for the degree

MPHIL IN SPACE STUDIES



September 19, 2019

SL19-05M

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

DECLARATION

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signature: ..... Signed by candidate .....

Date: ..... 2019-09-19 .....

# Queue Scheduling the Alan Cousins Telescope

Deneys Sean Maartens

## Abstract

The Alan Cousins Telescope is a 0.75-m automatic photoelectric telescope situated at the South African Astronomical Observatory, in Sutherland. The telescope was designed and built to execute a range of photometry programmes, but is used mainly for the long-term monitoring of variable stars. In addition, there is the potential for target-of-opportunity observations of unanticipated events, such as gamma ray bursts, and anticipated events such as occultations.

Ultimately the telescope is intended to be a fully robotic telescope with limited operational support needs. Some advance toward this goal has been made by a full hardware interface to allow queue executions of observations. The next phase is the implementation of an automated scheduler that will generate a queue of valid observations for each night of observation.

Queue scheduling algorithms are widely used in astronomy and the aim of this dissertation is to present a strawman scheduler that will generate the nightly observation queue. The main design of the scheduler is based on a merit-based system implemented at the STELLA robotic observatory, paired with the scheduling algorithms used by SOFIA.

The main drawback of the telescope is that it does not currently accommodate dynamically changing weather conditions. As a consequence, the main scheduling constraints are observation parameters, instrument ability, and for monitoring type observations, observation time window constraints.

## Acknowledgement

This research was supported by my full-time employer, the Southern African Large Telescope, who funded my studies and allowed me the time to work on it. The completion of this dissertation would not have been possible without the tireless aid of my wife, Ruby van Rooyen, who's math skills came to the rescue, and for proofreading multiple drafts of this document. In addition, I would like to thank my supervisor, Prof. P. Martinez for his guidance.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	General introduction to the problem . . . . .	2
1.1.1	Observing strategies . . . . .	3
1.1.2	Automated and robotic telescopes . . . . .	4
1.1.3	Alan Cousins telescope . . . . .	5
1.2	Defining the project focus . . . . .	9
1.2.1	Outline of the dissertation . . . . .	10
<b>2</b>	<b>Scheduling Parameters</b>	<b>11</b>
2.1	Astronomical scheduling considerations . . . . .	12
2.2	Astronomical scheduling parameters . . . . .	12
2.2.1	Celestial time and geographic location . . . . .	14
2.2.2	Target visibility . . . . .	16
2.2.3	Observing conditions . . . . .	16
2.2.4	Scientific priority . . . . .	18
2.3	Scheduler-specific terminology and definitions . . . . .	19
2.3.1	Timing requirements for observing programmes . . . . .	19
2.3.2	Ordering requirements . . . . .	22
2.3.3	Mechanical parameters . . . . .	22
2.4	Scheduler parameter space . . . . .	23
<b>3</b>	<b>Scheduler</b>	<b>36</b>
3.1	Basics of scheduling . . . . .	37

3.1.1	Planning . . . . .	37
3.1.2	Scheduling . . . . .	38
3.1.3	Observations . . . . .	39
3.2	The dispatch scheduler . . . . .	39
3.2.1	Astronomical veto functions . . . . .	41
3.2.2	Astronomical efficiency functions . . . . .	44
3.3	Putting it all together . . . . .	49
<b>4</b>	<b>Implementation</b>	<b>51</b>
4.1	Algorithm development . . . . .	52
4.2	Database . . . . .	55
<b>5</b>	<b>Testing and verification</b>	<b>61</b>
5.1	Test targets . . . . .	63
5.2	Basic functionality . . . . .	64
5.3	Observation scheduling . . . . .	73
<b>6</b>	<b>Summary and future work</b>	<b>79</b>

# List of Figures

1.1	Map of South Africa . . . . .	6
1.2	Photographs showing the APT . . . . .	6
1.3	APT system block design . . . . .	7
2.1	Equatorial coordinate system . . . . .	14
2.2	Elements of an observing programme . . . . .	20
2.3	LST, HA, and GHA . . . . .	25
3.1	Plane-parallel airmass model . . . . .	45
3.2	Target separation angle merit . . . . .	46
3.3	Target altitude merit . . . . .	47
3.4	Target rise time merit . . . . .	48
3.5	Target observation window merit . . . . .	49
4.1	Initial database logical model . . . . .	57
4.2	Database logical model . . . . .	59
5.1	Observation schedule for sequential targets . . . . .	66
5.2	Sequential target culmination queue verification . . . . .	67
5.3	Improved queue generation for sequential targets . . . . .	68
5.4	Single target observed repeatedly . . . . .	69
5.5	Priority merit implementation . . . . .	70
5.6	Targets overlapping at culmination time . . . . .	71
5.7	Different priority targets overlapping at culmination time . . . . .	71
5.8	Incidental condition: target observation paused . . . . .	72

5.9	Updating observational conditions . . . . .	72
5.10	Observations optimised using the culmination merit . . . . .	72
5.11	Observations optimised using the airmass merit . . . . .	72
5.12	Skyview showing culmination scheduling . . . . .	74
5.13	Skyview showing airmass scheduling . . . . .	74
5.14	Observation night over culmination . . . . .	74
5.15	Observation night over airmass . . . . .	74
5.16	Queue evaluation scores . . . . .	75
5.17	Observation night over airmass . . . . .	75
5.18	Multi-target, multi-merit full night schedule . . . . .	76
5.19	Summer solstice queue . . . . .	77
5.20	Winter solstice queue . . . . .	77
5.21	Queue permutations evaluated for summer . . . . .	78

# List of Algorithms

3.1	Function Veto(visible) . . . . .	42
3.2	Function Veto(magnitude) . . . . .	42
3.3	Function Veto(sky brightness) . . . . .	43
3.4	Function Efficiency(target altitude) . . . . .	47
4.1	StartTime . . . . .	53
4.2	ForwardPlan . . . . .	54
4.3	LookAhead . . . . .	55

# List of Acronyms

<b>ACT</b>	Alan Cousins Telescope . . . . .	ii, 5, 7, 10, 13, 33, 50, 79, 80
<b>AI</b>	artificial intelligence . . . . .	4, 5, 9
<b>API</b>	application programming interface . . . . .	56
<b>APT</b>	Automatic Photometric Telescope . . . . .	vi, 5–7
<b>DCSP</b>	dynamic constraint satisfaction problem . . . . .	10, 51
<b>Dec</b>	declination . . . . .	14, 15, 24–26, 64
<b>EAV</b>	entity-attribute-value . . . . .	58, 59
<b>FWHM</b>	full width at half-maximum . . . . .	18
<b>GHA</b>	Greenwich hour angle . . . . .	vi, 24, 25
<b>GPS</b>	Global Positioning System . . . . .	5, 7
<b>HA</b>	hour angle . . . . .	vi, 15, 16, 24–26, 65, 68
<b>IT</b>	information technology . . . . .	35
<b>JSON</b>	JavaScript Object Notation . . . . .	59, 60
<b>LHA</b>	local hour angle . . . . .	24, 26
<b>LST</b>	local sidereal time . . . . .	vi, 15, 16, 21, 24, 25
<b>ML</b>	machine learning . . . . .	4, 9
<b>PC</b>	personal computer . . . . .	5, 7
<b>PLC</b>	programmable logic controller . . . . .	5, 7

<i>LIST OF ACRONYMS</i>	x
<b>PMT</b> photomultiplier tube . . . . .	5, 7
<b>PSF</b> point spread function . . . . .	18
<b>PSU</b> power supply unit . . . . .	7
<b>PWV</b> precipitable water vapour . . . . .	18
<b>RA</b> right ascension . . . . .	14, 15, 24–26, 64
<b>SAAO</b> South African Astronomical Observatory . . . . .	5, 7, 8, 64
<b>SHA</b> sidereal hour angle . . . . .	24
<b>SNR</b> signal-to-noise ratio . . . . .	12, 19
<b>SOFIA</b> Stratospheric Observatory for Infrared Astronomy . . . . .	ii, 10, 51, 52
<b>STELLA</b> STELLar Activity . . . . .	ii, 2
<b>TAC</b> time allocation committee . . . . .	18, 19, 21, 31, 62, 68
<b>TCS</b> telescope control system . . . . .	13, 29, 33
<b>TLE</b> two-line element . . . . .	32
<b>TOMs</b> Target and Observation Managers . . . . .	56
<b>TOO</b> target-of-opportunity . . . . .	ii, 8, 12, 18, 19, 21, 38, 40
<b>UTC</b> Coordinated Universal Time <sup>1</sup> . . . . .	23, 24
<b>XML</b> Extensible Markup Language . . . . .	59

---

<sup>1</sup>This incongruous term has its origins in a compromise by the International Telecommunication Union to enable a universal abbreviation for use in all languages. (National Institute of Standards and Technology, 2016)

# Chapter 1

## Introduction

Scheduling related to science instruments is typically more complex and quite different from standard, application-related scheduling problems which are routinely solved in industry. This has largely to do with the uncertain exploratory nature of science. Scheduling the use of scientific instruments requires making choices that impact other choices later, and involves many interacting complex constraints over both discrete and continuous variables. Furthermore, sets of constraints are dependent on a given science project, while new types of constraints may be added as the fundamental problem changes (Frank, 2000).

The scheduling and acquisition of astronomy data is a multi-objective problem and can be broken into four sections: *a)* planning, *b)* scheduling, *c)* observing and *d)* data management (Denny, 2004). The first three sections represent a strong interdependent problem generally referred to as *scheduling*. Data management is only loosely related and feeds back into planning and scheduling through the contribution of previously acquired data to the assessment of progress toward the achievement of scientific goals and the scheduling of further observations to meet those goals.

Astronomy projects are complex, often consisting of inseparably connected constraints, requiring long-term planning as well as short-term optimisation. For observatories this translates into telescope scheduling that focuses on optimising resource utilisation as

the goal, while contending with the reality that sky conditions can change significantly during an observing session, thereby causing breakage in pre-prepared queue schedules. In addition, rapidly changing scientific priorities may require prompt, unplanned observations. These issues give rise to the need for a scheduling system that is capable of recovering from periods of bad observational conditions, accommodating changing priorities, and integrating newly added observations during operation (Denny, 2004).

The challenge is to optimise the scientific return while maintaining good scheduler etiquette. There are three criteria for a *good* schedule: *a*) fairness, *b*) efficiency, and *c*) sensibility. A fair schedule balances time allocations between users such that they all share *good* and *bad* observing times equitably. An efficient schedule is one that maximises instrument utilisation and strives to match observations with required conditions. A sensible schedule is one that attempts only those observations that are possible under the current observing conditions (Denny, 2004).

This dissertation presents a strawman scheduler that makes use of a dynamic queue of observations. Using the merit implementation presented by Granzer (2004) which is used to approach the scheduling problem at STELLA, paired with the algorithm presented by Frank and Kürklü (2003), the scheduler allows for the dynamic conditions during execution of an observation and best-choice selection based on available observation plans (Wall, 1996).

## 1.1 General introduction to the problem

Observing time is a scarce resource (Johnston, 1988a) which is subject to the vagaries of the weather. Fortunately not all astronomical observations require the very best atmospheric conditions, hence the need for planning and scheduling to take full advantage of the variations of the weather conditions (Gómez de Castro and Yáñez, 2003).

The ultimate goal of scheduling is to maximise the scientific impact of the telescope. It can be argued that the following goals contribute the science impact (Colomé et al.,

2012):

- Minimising the telescope idle time;
- Minimising the time overheads due to the scheduler—in the case of dynamic scheduling;
- Maximising the time available for science observations;
- Maximising observations of the highest scientific priority; and
- Maximising the quality of the collected data, i.e. matching the observations' execution constraints to the execution conditions.

From the goals above, it follows that scheduling of astronomical observations is an example of a multi-objective problem, where different factors must be optimised.

This requirement for planning and scheduling is applicable in a wide range of sectors, from the chemical, petrochemical, and pharmaceutical industries, to waste management (Verderame et al., 2010). It falls into the NP-hard class of problems (Gómez de Castro and Yáñez, 2003), where it is computationally infeasible to enumerate all of the possible permutations in order to select the optimal solution (Johnston, 1989); only a reasonable approximation of the optimal solution can be reached in a finite time.

### 1.1.1 Observing strategies

Most astronomical observatories employ one or more of the following modes of observation. In the *classical* mode of observing, an astronomer travels to the telescope for a predetermined length of time—typically in the order of one or more weeks—to observe their own targets for the duration of their allotted time. An alternate to in situ observing is *remote observing*, carried out by the astronomer via remote control from a site more convenient than the telescope itself. A refinement of the traditional observing mode is *service observing*, where on-site observing staff perform the observations based on specifications prepared by the astronomer. It is noteworthy that service

observing introduces the possibility to conduct multiple observing programmes concurrently (Johnston, 1988b). Service observing may itself be further refined into *automated observing*, where the astronomer prepares the specifications of the observation and submits it for execution. However, instead of on-site staff performing the observation, an automated telescope performs all the steps necessary to complete the observation. As an even further refinement, *robotic* observing is performed by telescopes which operate autonomously and use advanced artificial intelligence (AI) and machine learning (ML) algorithms to select and schedule observations without any human interaction.

### 1.1.2 Automated and robotic telescopes

The automation of telescopes only really became a possibility with the advent of the microcomputer (Genet, 2011). Before the microcomputer was introduced, a few attempts at automating telescopes by controlling them remotely with a mainframe were made in the 1950s and 1960s, but this mostly proved to be unreliable (Castro-Tirado, 2010). In contrast, remote observing only became practicable with advances in network and communication technologies in the late-1980s (Bresina et al., 1994), or in other words, with the advent of the Internet.

The evolution to telescope automation from remote to robotic can conceptually be summarised in the following categories—adapted from Castro-Tirado (2010):

**Remotely operated telescope** A telescope system that performs remote observations following the request of an observer. Thus the observer instructs the remote telescope to perform various actions which are then subsequently carried out by the telescope.

**Queue-scheduled automated telescope** A telescope that performs queued observations, without the immediate help of an observer. The astronomer acts mostly in a supervisory role to react to schedule breakage or controlling incidental unscheduled observations.

**Autonomous observatory** A telescope that performs remote observations and is able to adapt itself to changes in observing conditions or priorities during the task execution without any human assistance. This requires a sufficient level of situational awareness by the telescope, especially with regard to weather conditions.

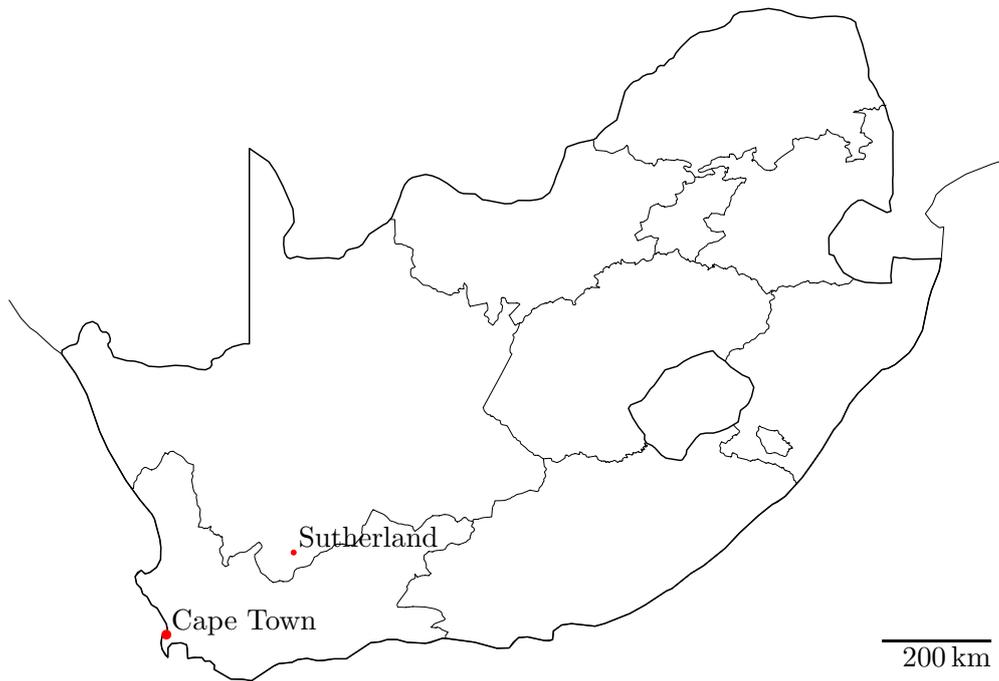
**Intelligent observatory** A robotic observatory in which decisions are taken by an AI system. This implies that the telescope is in full control of selecting the optimum target to observe from the list of candidate targets.

Each level of automation builds upon the previous level; it is not possible to have a robotic intelligent observatory, for example, without the automation of all of the requisite components, or without the input from a weather monitoring system of some kind.

### 1.1.3 Alan Cousins telescope

Queue scheduling algorithms are widely used in astronomy (Mora and Solar, 2010) and we will use this approach to schedule the Alan Cousins Telescope (ACT) which is situated at the South African Astronomical Observatory (SAAO), in Sutherland. The ACT, sometimes referred to as the Automatic Photometric Telescope (APT), is a 0.75-m automatic photoelectric telescope commissioned in the mid-2000's (Martinez et al., 2002).

When it was originally commissioned, the ACT control system ran on two MSDOS 6.1 personal computers (PCs). In 2010 the main control elements of the ACT were upgraded (Van Heerden, 2011) and now consists of a Linux based PC and a programmable logic controller (PLC). The PC contains cards that connect to the photomultiplier tube (PMT), acquisition system, telescope drives and the time system, and communicates with the PLC. Time is obtained from Global Positioning System (GPS) signals. The PLC controls the rest of the telescope functions such as the telescope focus, acquisition mirror motions, the filter wheels and the aperture wheel. Figure 1.3 shows a schematic

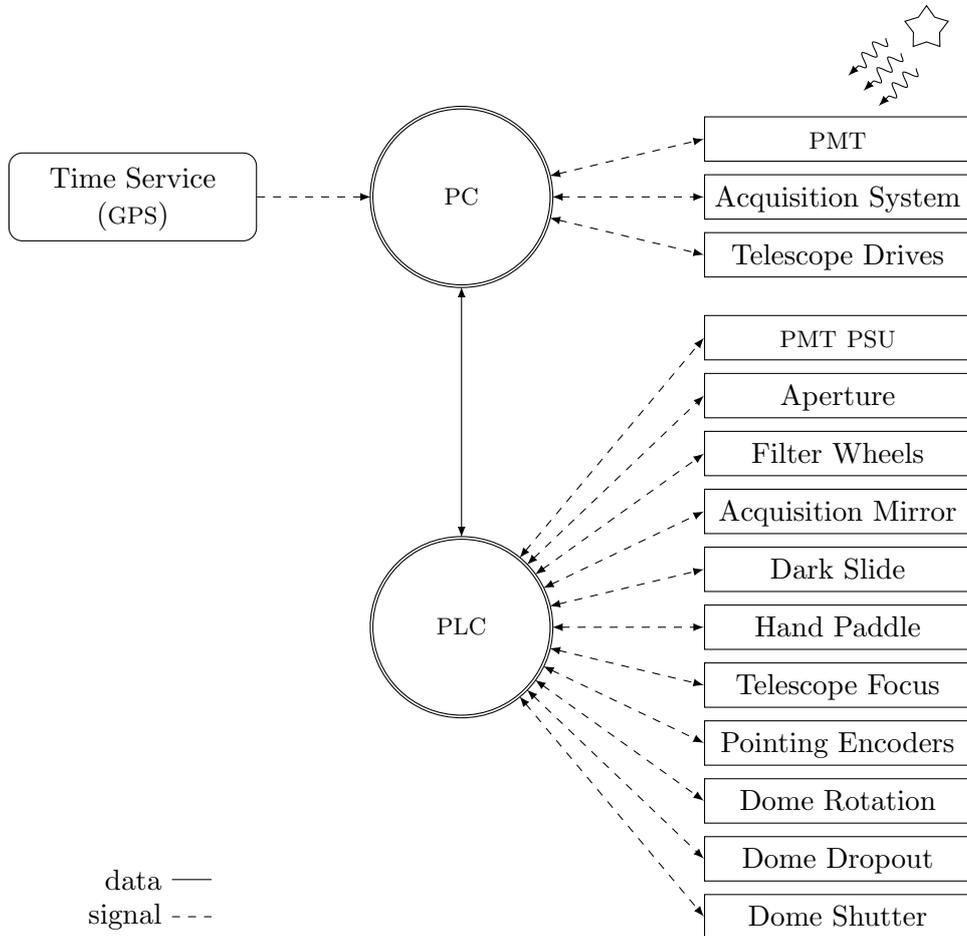


**Figure 1.1:** Map of South Africa, indicating the locations of Cape Town and Sutherland.



**Figure 1.2:** Photographs of the Automatic Photometric Telescope (APT), showing the dome housing the telescope (left), the telescope itself (centre), and some detail of the base of the telescope (right).

outline of the telescope system.



**Figure 1.3:** Automatic Photometric Telescope system block design (Van Heerden, 2011).

Currently, the ACT does not have dedicated weather sensors. For weather information, it relies on data obtained from other facilities at the SAAO in Sutherland, that publish their data either on the local intranet, or globally via the Internet.

The telescope system hibernates during daytime; at sunset the control computer opens the dome and performs the telescope’s initialisation routines. Then the telescope begins to work its way through a pre-prepared list of observing targets. It steps through this list and for each target the telescope acquires the target using a pattern matching algorithm. It then proceeds to perform observations of the target. During the night’s observation, a persistent pointing error may cause the system to re-initialise itself; if pointing cannot be reinitialised successfully—perhaps due to cloud obscuration of

targets—the system shuts itself down and closes the dome to avoid damage to the telescope. To protect the system, software and hardware limits restrict the movement of the telescope, protecting it from collisions with objects in the dome or from trying to reach unobtainable pointing positions. The system shuts down again at morning astronomical twilight. The observational data is recorded locally and transferred to the SAAO offices in Cape Town for analysis during the course of the day.

Ultimately the telescope is intended to be a fully robotic telescope with limited operational support needs. Some advance toward this goal has been made by a full hardware interface to allow automated queue executions of observations (Van Heerden, 2011). The next phase is the implementation of an automated scheduler that will generate a queue of valid observations for each night of observation.

The telescope was designed and built to execute a range of photometry programmes, but is used mainly for the long-term monitoring of variable stars. In addition, there is the potential for target-of-opportunity (TOO) observations in the form of unanticipated events, such as gamma ray bursts, and anticipated events such as occultations. All photometric observations are restricted by the requirement of clear skies. The current observation strategy is very much up to the assistant astronomer, who evaluates whether prevailing conditions are good enough to start observations, as well as when to end observations if the weather conditions deteriorate beyond what is needed for photometric observations.

The execution of schedule blocks cannot accommodate dynamically changing weather conditions, except for evaluating predicted conditions when generating the scheduling queue in the planning phase. As a consequence, deciding whether the instrument will observe, or not, is considered to be outside the control-and-monitoring functionality of the system and is therefore not used during planning. The main scheduling constraints are thus observation parameters, instrument ability, and for monitoring type observations, observation time window constraints.

## 1.2 Defining the project focus

Selecting programs suitable to be observed from one observation run to the next can be as simple as an ordered list of observation blocks, using queue scheduling algorithms. On the other hand, it may require some more intelligence: selecting from multiple, overlapping choices requiring Markov decision processes (Littman and Majercik, 1997), or harnessing Genetic algorithms (Wall, 1996) to build suitable sets of queues with the potential combinations of environmental and atmospheric conditions.

Various methods exist to select the *best* possible candidate in the multitude of possible solutions of the challenging problem of observation scheduling. The main distinction that can be made between the methods is between exact methods and heuristics (Buchner, 2011); exact methods include linear programming and constraint satisfaction problems, while heuristics include various AI approaches, simulated annealing and evolutionary algorithms.

Dynamic/linear programming methods are greedy, brute-force techniques that are computationally expensive, but these methods have the advantage of *always* finding a global optimal solution (Buchner, 2011). Computational performance can be improved by selective and simplifying assumptions that are valid to the application, such as the case with dynamic constraint satisfaction problem, and constraint optimisation problem methods (Frank and Kürklü, 2003).

Heuristic methods fall in the category of AI algorithms, which include ML, simulated annealing, and evolutionary algorithms (Buchner, 2011). All these algorithms require training and upfront knowledge in order to make correct decisions. This input generally comes from models generated by the observatory and is bootstrapped into the scheduling strategy. The Hubble Space Telescope (Johnston and Adorf, 1992) and the Liverpool Telescope are prime examples of such systems. The Liverpool Telescope is notable in that it is currently the world's largest fully robotic terrestrial telescope. It specialises in time domain astrophysics and has a dedicated instrument suite giving imaging, spectroscopic and polarimetric capabilities. (Fraser and Steele, 2004)

The development and implementation of such extensive schedulers takes a lot of time and experience and is not fundamentally required to implement automated operation of the small ACT. For smaller telescopes the much more agile dynamic scheduling strategies have been shown to work very well.

An appropriate scheduling strategy for the ACT is *dispatch*, or *just-in-time*, scheduling. While the scheduler fundamentally makes the best choice for the next observation it also implements simple models to pro-actively make the nominal schedule more robust. This approach is able to adapt to changing conditions, new requests, and acquisition errors, while still maintaining a reasonable measure of efficiency (Denny, 2004; Granzer, 2004).

The ACT queue scheduler will combine the dynamic dispatch scheduling strategy with optimisation using the dynamic constraint satisfaction problem optimisation (Frank and Kürklü, 2003)—implemented by the Stratospheric Observatory for Infrared Astronomy (SOFIA).

### 1.2.1 Outline of the dissertation

Chapter 2 introduces the reader to the scheduling problem by providing definitions for the variables generally playing a role in the scheduling of telescopes.

Chapter 3 discusses the merits and constraints implemented for the ACT queue scheduler.

Chapter 4 and Chapter 5 present the strawman scheduler’s implementation and verification, developed in Python, in some technical detail.

Chapter 6 rounds off the presentation of the strawman scheduler by summarising the scheduling technique implemented, as well as some discussion on generalising the implementation in future work.

## Chapter 2

# Scheduling astronomical observations

The main aim behind any automated scheduling strategy is to optimise telescope time usage and scientific productivity. The scheduler must take into account a number of hard and soft constraints, and use some optimisation strategy to generate a series of observations that can be scheduled. Furthermore, during operations the scheduler must continue to evaluate these constraints in order to timeously identify which observations to execute, while ignoring observations that cannot be performed (e.g. due to technical or scientific reasons), as well as handling interruptions (e.g. due to weather) and resuming observations when possible.

This is achieved by identifying constraints, derived from the proposal's observational requirements as a mathematical relation between some dependent and independent parameters. For the most part, constraint quantifiers make up the building blocks of the scheduler. These quantifiers, in turn, define the science-specific parameter space used to evaluate the observational productivity.

Operational parameters may be general to astronomy or unique to an observatory/instrument/observation mode. Other influences will depend on observatory policies and procedures, such as those related to long-term projects, or compensations for time

lost due to TOO or other reasons.

## 2.1 Astronomical scheduling considerations

A typical observation request provides the name, coordinates and brightness, for the objects to be observed. In addition, observational information may be included such as the type of data required, the required signal-to-noise ratio (SNR), the amount of time requested, the relative importance of the observation<sup>1</sup>, and a set of constraints on the observation (Frank, 2000). The requesting astronomer may specify constraints explicitly given the observation request. The observatory must add the implicit and instrument/telescope specific constraints.

Astronomical observations are regulated by a wide range of parameters; some predictable, such as target visibility or mechanical constraints, some unpredictable, such as weather conditions or telescope/instrument failure, while others are defined by observatory policies. Additionally, the constraints may vary depending on instrument and observation type. This chapter describes the parameter space defined by the various constraints for the scheduling algorithm.

## 2.2 Astronomical scheduling parameters

The two major drivers for scheduling astronomy observations are time and the observing conditions.

Astronomy observations are very much dependent on favourable weather conditions. While predicted future conditions can be employed to some extent in planning and scheduling, the scheduler is very much dependent on knowledge of the current weather affecting the operational dynamic observing phase. The schedule queue will contain all viable observations and, at the time of execution, the measured atmospheric conditions

---

<sup>1</sup>A prioritisation weighting assigned by a time allocation process that reviews all telescope time requests.

will dictate which observations can be performed. In addition, many telescopes have weather stations that are linked to the telescope control system (TCS), with the ability to shut down the telescope in adverse weather conditions. These situations will interrupt the observation and the TCS will not allow further observations to be scheduled until weather conditions improve. This is an interruption event and the scheduler will simply continue—where possible—after the interruption.

Since the ACT does not have a TCS interface to an associated weather station it is very much up to the remote assistant to evaluate if conditions are good enough to start observations, as well as to end observations when the weather conditions deteriorate beyond what is required for photometric observations. The main scheduling constraints are thus observation parameters, instrument ability and observation time window constraints.

Time is the second most important aspect when planning and setting up the observation queue. There exist various measures of time in astronomy, with time representation formats even being used to represent directions and separation of objects in the sky as angular values. The standard assumed format is degrees/hours, arcminutes and arcseconds.

In general time variables are expressed in:

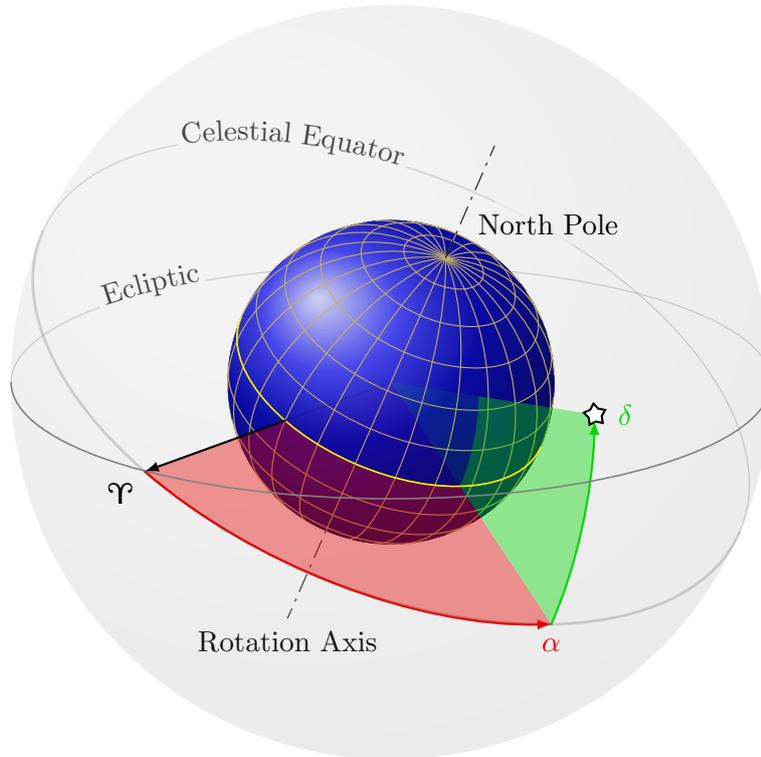
years — integer value,  
 months — 1–12,  
 day of the month — 1–28..31,  
 hours — 0–23,  
 minutes — 0–59,  
 seconds — 0–60<sup>2</sup>

and, if fine time resolution is required, fractional seconds up to the milli-, micro-, or even nanosecond scale. Various subsets of these parameters are used depending on the time variable expression.

---

<sup>2</sup>The upper-limit of 60 is due to the occurrence of leap seconds.

## 2.2.1 Celestial time and geographic location



**Figure 2.1:** Equatorial coordinate system, showing the celestial sphere with the Vernal Equinox ( $\Upsilon$ ), indicating where the Sun crosses the Earth’s equatorial plane from South to North, defining the zero value of *right ascension* ( $\alpha$ ), the East-West position of a star in the sky. Also shown is the *declination* ( $\delta$ ) which defines the position above or below the equatorial plane.

Astronomical targets are referenced using a celestial coordinate system. For simplicity, this dissertation will specifically use the equatorial coordinate system to represent celestial targets. The equatorial coordinate system defines the origin of the coordinate system to be the centre of the Earth. It also projects the Earth’s equator outwards to form the celestial equator on the celestial sphere, as illustrated in Figure 2.1.

The location of a target on the celestial sphere is specified in right ascension (RA) and declination (Dec), while terrestrial telescope and horizon mask information are given using horizontal coordinates. The latter consists of *azimuth*, measured clockwise from North, and *elevation*, measured above the horizon. These two coordinate systems are

related through:

$$\sin(a) = \sin(\phi_o) \sin(\delta) + \cos(\phi_o) \cos(\delta) \cos(h)$$

where  $a$  is the target elevation angle,  $\phi_o$  is the telescope latitude,  $\delta$  the target object's declination, and  $h$  is the hour angle (HA), that is, the difference between the local sidereal time (LST),  $\theta_L$ , and the object's right ascension,  $\alpha$ :

$$h = \theta_L - \alpha.$$

In other words, HA indicates the amount of time until (−ve) or since (+ve) a given object's passage across the local meridian.

To use the time-based HA in trigonometric functions, an angular value is required. As there are  $360^\circ$  in 24 hours—multiply the hour value by  $360^\circ/24 \text{ h} = 15^\circ/\text{h}$  to obtain degrees.

The azimuth angle,  $A$ , can be determined by using either

$$\sin(A) = -\frac{\sin(h) \cos(\delta)}{\cos(a)},$$

or

$$\cos(A) = \frac{\sin(\delta) - \sin(\phi_o) \sin(a)}{\cos(\phi_o) \cos(a)},$$

where  $h$  is the HA,  $\delta$  is the target declination,  $\phi_o$  is the telescope's latitude, and  $a$  is the target elevation. As the Earth orbits the Sun during the course of the year, the sky—as observed from the Earth—rotates and with it the HA range relative to some local time standard.

It is worth noting that the equatorial coordinates, right ascension (RA) and declination (Dec), should be precessed and integrated to the epoch assumed by the telescope pointing calculation—the current standard reference epoch is J2000 which is (approximately) on January 1, 2000 at noon UTC.

### 2.2.2 Target visibility

In order to observe a target, it must be in a part of sky visible to the telescope at some time during the observation period. The threshold for target visibility can be derived in terms of a minimum altitude angle, or alternatively as an hour angle (HA) relative to local sidereal time (LST).

The variable against which the target rise/set ( $T_0$ ) threshold is evaluated is the minimal altitude angle defined as the local horizon angle in an azimuth direction. The definition of local horizon is intentionally broad and may include instrumental limits, local topographic features or man-made obstructions that collectively define the telescope pointing constraints as a function of azimuth—also known as a *horizon mask*.

A target is considered visible over the time period that the target elevation is above the rise/set threshold values. This period is defined as the target visibility period ( $T_{vis}$ ). For observation programmes with multiple targets the time measure parameters must be defined relative to the altitude of a selected target or reference position and the target visibility period sufficiently long to ensure observation of all required targets.

Additional horizon angle limits may be imposed to ensure favourable image quality for calibration or correction procedures. Atmospheric absorption at lower elevations may cause degraded results, such as reduced intensity, due to spectral absorption and scattering. The effect of atmospheric dispersion is generally specified using some observatory airmass model,  $z(h)$ .

### 2.2.3 Observing conditions

Depending on the type of observation and the scientific objectives, additional constraints may accompany an observation request.

Astronomers may provide explicit constraints on particular observations to ensure that the data are of adequate quality for their programme requirement. For example, the astronomer may require that the object be sufficiently separated on the sky from solar

system bodies, place a constraint on the lunar phase or acceptable sky brightness, or that airmass should be below a certain threshold. These constraints also dictate when a target may be observed (Frank, 2000).

A nearby or bright Moon can affect optical observations in several ways; Moonglow can increase the noise floor of an image when the Moon is close to the field of view, making the calibration difficult or it may even cause reflections or stray light within the telescope optics.

Rules around lunar constraints—such as lunar phase, lunar altitude and minimum lunar separation angle—create observational zones of avoidance. Other zones of avoidance may be defined for bright satellites, other planets, etc., depending on the exact timing and celestial coordinates of an observation.

In addition, atmospheric conditions related to sky transparency constraints such as photometric conditions must be clearly specified. Examples of such constraints can be:

- a) *Photometric conditions* assessed by analysing data from photometric standard stars—no visible clouds, transparency variations under 2%—and
- b) *Spectroscopic conditions* when less than 10% of the sky (above 30 degrees elevation) is covered in clouds, and transparency variations are under 10%.

Weather forecasts, or known weather patterns, can be used to predict rain or cloudy conditions; while humidity and wind conditions could be tracked during the nightly observation run to make adjustments to the schedule in near real time.

For some observations excellent atmospheric conditions are very important. To schedule these observations it may be important to continuously evaluate dynamic atmospheric conditions—these types of observations may be difficult to preschedule.

Cloudy conditions can cause focus and guiding failures in optical observations since the telescope may not be able to find the guide star or the star may be obscured by clouds during long exposures and thus the image quality could be affected. Weather

conditions need to be tracked during the night and observations adjusted accordingly.

Other parameters sensitive to atmospheric conditions are seeing and precipitable water vapour (PWV). Seeing is defined as the full width at half-maximum (FWHM) of the point spread function (PSF) of a point-like source image in arcseconds, at the wavelength of observation. Thus, it is an indication of the the image quality obtained through the atmosphere, telescope and instrument. When the local atmosphere above the observation site is unstable or turbulent this can result in *soft* focus and shaky guide stars.

Acceptable upper limits for the precipitable water vapour (PWV) may be specified in proposals for some instruments or observations. These values can generally be obtained from planning tools and other sources of information available to the telescope.

#### 2.2.4 Scientific priority

Priority for science observations is determined by the time allocation committee (TAC) and can be any agreed method of indicating priority such as *low*, *medium*, and *high*. In exceptional situations, a separate higher priority class such as *urgent* can be assigned, but it is not generally used. Urgent priority observations may include rare condition programmes which are expected to have a high scientific impact but can only use observations acquired under exceptional operational conditions such as very low precipitable water vapour (PWV) and superb seeing. TOO observations are considered independent and can be assigned a priority such as *must-run* or *non-scorable*, in preference to having an additional operational tag.

Quite often it is necessary to observe TOO objects for a short period of time before and after the initial event and the *high/urgent* priority is needed to ensure observation at the specified cadence (nightly, etc.) for the duration of a particular observing programme. These follow-up and confirmation observations may be required to have gone through the TAC process or could be communicated directly with the staff astronomer, depending on scientific impact.

TOO observations are evaluated simply based on position, environmental conditions and are then executed. Should the TOO observation fail or the prevailing conditions make the observation impossible, the observation is classified as *failed* and the next observation is executed. Rescheduling of failed TOO observations is defined by observatory policies. Note should be made of the underlying inherent assumption here: it is highly unlikely that multiple TOO alerts will be received simultaneously; therefore the scheduler will handle this on a first-come, first-served basis, unless weighted differently by the observatory.

## 2.3 Scheduler-specific terminology and definitions

The default state of the instrument is always assumed to be operational. Extra information in the form of programme tags can be used to indicate the state of the observation programme, such as *active*, *paused*, *completed*, or *suspended*. Active programmes can be scheduled for observation and include triggered TOO events. Programmes that have reached some science goal such as SNR, or the targeted observational data or observation time limit are considered as *completed*. Depending on policies and procedures, programmes that do not have enough allocated time remaining to qualify for another full observation may be marked as either *completed* or *paused*. Furthermore, programmes may be *paused* at any time by the observer, TAC or observatory, also dependent on policy and procedures.

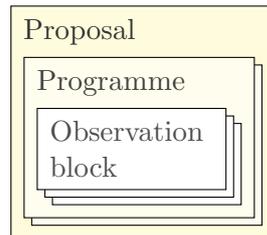
### 2.3.1 Timing requirements for observing programmes

Although, at its heart, telescope scheduling is all about making effective use of telescope time, different observing projects have different sorts of observing time requirements. In this section we describe the various timing requirements typically encountered in observing programmes.

Each project proposal, when accepted, will require observations over a period of time

defined as the total allocated time. This allocated time can either be actual observation time in hours, or until a given scientific criterion is met. Either option can translate into enough time for a once-off observation, or multiple observations over an extended period. The minimum observation time requirement ensures that the observation time is long enough to provide sufficient coverage, sensitivity or resolution to produce usable data that can be calibrated and/or combined to produce the resultant science data of adequate duration and quality to meet the required scientific goals.

As graphically illustrated in Figure 2.2, accepted proposals consists of one or more observation programme(s) that may contain a single observation block or a set of observation blocks. An observation block is a unit fully describing an observation, each with specific instrument setup and time or science requirements. It is the observation blocks that will be used by the scheduler and is henceforth simply referred to as an observation.



**Figure 2.2:** Schematic depiction of the elements of an observing programme proposal.

An observation may contain a single pointing (target-calibrator or target-comparator pair), or a list of target and comparator/calibrator sources. In relation to the minimum observation time block, a set of pointings could have the requirement to be scheduled as a unit, and will be referred to as a group.

The observation group information is used to generate the observation block for execution and constraints must be satisfied for all sources in the group. A group's length may represent observational data acquisition time only and exclude startup, shutdown and instrument calibration time, or may include overhead, either depending on observatory policies, or observation type.

Splitting up an observation programme into groups can require special timing condi-

tions, such as cadence, to be attached to the request. How the observation time will be utilised depends on the scientific goals of a particular project. For example, monitoring variable stars requires periodic observations over some period of time, which may be hours/days/years. Observations and follow-ups of TOOs or transient events may be limited to a small window of opportunity.

The scheduler must take TOO and other transient events into account and prioritise the observation of these targets during the appropriate observation window. These observations are time-critical and must be executed at the time specified, or as close to that time as possible, except where observational or mechanical constraints prevent this from happening.

Monitoring observations are considered time-constrained observations. Additional variables for monitoring may include the cadence or sampling frequency (with time window constraints) to obtain data spread over a timespan of interest, or even a requirement to start at either the same local sidereal time or local standard time. For some observations the cadence does not matter and the only requirement is some observational data with a reasonable spread over the allocated time.

Unless assigned differently by the TAC, all observation block groups in a observation programme will share the priority assigned to the programme and will thus have an inherent selection priority conflict. Weights can also be used within a programme that contains multiple groups to give precedence to selected groups when evaluating the programme during scheduling. The priority of a project is assigned during the proposal evaluation phase. Once assigned, programme priorities do not change—unless the proposal is re-evaluated. Hence, to not penalise lower priority programmes and to allow a fairer scheduling schema, weighting of programmes may be employed:

- As programmes move closer to completion they may be assigned higher weights, thus favouring a strategy to complete observations and get the programme off the list of active programmes;
- Programmes may be weighted inversely to completeness to favour collecting data

for as many projects as possible;

- Default: a natural weighting where only scientific priority and observational constraints influence observations.

### 2.3.2 Ordering requirements

Some observations may have explicit constraints restricting the order in which they are to be performed. In addition, instruments may need to be calibrated by observing particular standard/calibration objects before, and also after, the primary observation of interest is performed. The telescope may need to be tuned/focused/calibrated at the beginning and periodically during the night by observing objects with particular characteristics. High-precision tuning or calibration may require observing the same object at multiple elevations, for instance. These requirements impose ordering constraints on the observations that must be obeyed (Frank, 2000).

### 2.3.3 Mechanical parameters

Time used for non-programme observational events, such as startup, shutdown and all intermittent system calibrations, is considered dead time and reduces the time available to observe scientific target objects.

Maintenance cycles are generally scheduled in advance and can be included in the observational scheduling during planning. In general, maintenance cycles are planned well in advance since the telescope is effectively non-operational during that time. For optical telescopes, most maintenance is carried out during daytime, but daytime maintenance delays, or unplanned events, can prevent the telescope from returning to an operational state. These can be handled by simply locking the system, thereby preventing the start of routine operations.

Technical failures are generally unpredictable and may interrupt observations at any

time. These may include mechanical, electronic and software failures<sup>3</sup>.

Operational overheads cause dead time that must be taken into consideration during scheduling. Telescope structures take a finite amount of time to slew to a designated target coordinate. This slew time can be calibrated and generally adds a small amount of extra procedural time to the group observation time allocated. When targets are widely separated on the sky this slew time has a negative impact on the telescope performance. It is thus preferable for the scheduler to try to cluster targets together in sky regions to minimise slew time. This behaviour can be really advantageous for highly oversubscribed instruments.

Additionally, if the system overhead, such as readout and instrument setup time, is significant compared to the observation setup time, it must also be taken into account. This can be done by allotting some fixed delay between scheduled observations.

## 2.4 Scheduler parameter space

Inspecting the presented observational constraint descriptions allows for the identification of relevant parameters, as well as the relations between them that leads to the constraints.

This section summarises the respective parameter definitions, which will be assumed in the next chapter, when the relation between them are used to identify constraints and restrictions for the implementation.

### Time Measures

#### UTC

To avoid complexities introduced by different time zones and local time corrections such as daylight saving, most telescopes use Coordinated Universal

---

<sup>3</sup>In a conventional (i.e. non-robotic) telescope, an experienced human astronomer may find a “work-around” for the fault in order to continue the observations. Robotic telescopes are generally less resilient in this regard—at least for now!

Time (UTC) as the standard time unit. The UTC standard forms the basis for the world's civil time.

### **LST**

Sidereal Time is a time scale based on the Earth's rotation relative to the stars, rather than the Sun. Local Sidereal Time (LST) is Sidereal Time adjusted for geographic location.

### **HA/LHA/GHA**

Hour Angle (HA), Local Hour Angle (LHA) and Greenwich Hour Angle (GHA) essentially indicates how long it has been since the object last passed a given meridian. For GHA, the prime meridian is used, and for LHA the convention is to use the local meridian. The valid range for an HA is the meridian in question  $\pm 12$  sidereal hours. Somewhat less frequently used is the Sidereal Hour Angle (SHA), which uses the Vernal Equinox—the first point of Aries—as reference.

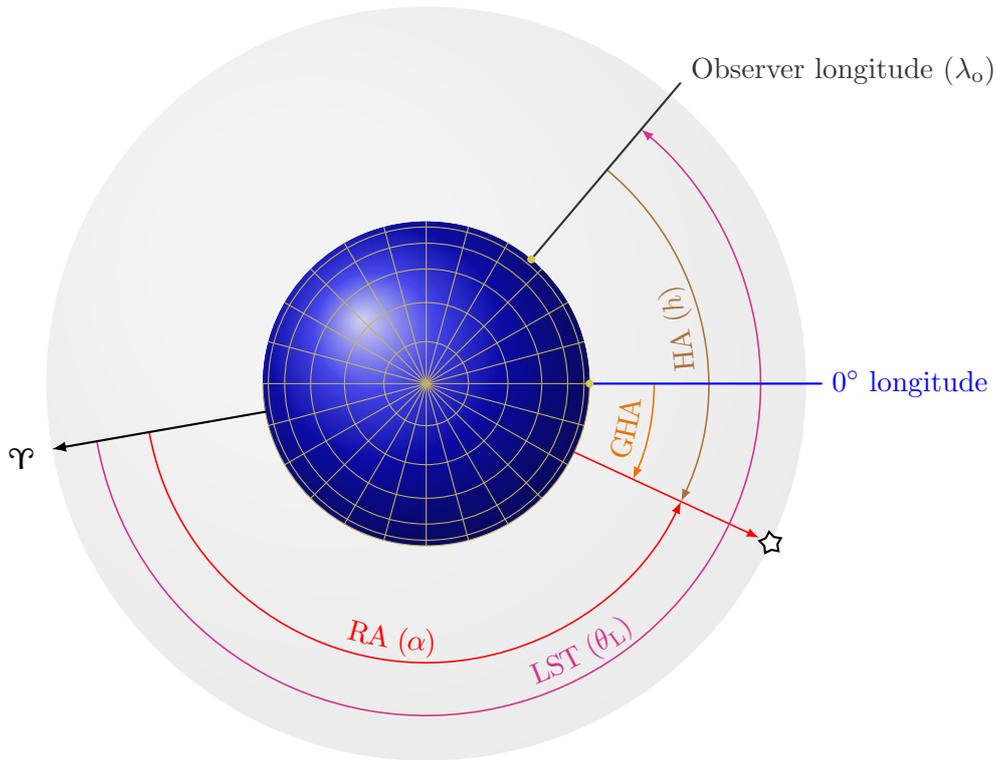
### **Epoch**

This parameter serves as a reference time for the (RA, Dec) pair. The celestial coordinates of a given target object are time varying due to, among other factors, precession of the equinoxes. The reference date for a given coordinate is called the Epoch. Precession of the Earth's rotational axis, caused predominantly by gravitational effects of the Moon and the Sun on the Earth's equatorial bulge, cause celestial coordinates to drift over time. To account for this drift, coordinates are specified with respect to a reference point in time. The current commonly used reference Epoch is J2000. Coordinates may need to be precessed to the current date to aid accurate pointing.

## **Geographical variables**

### **Geographic location**

The geographic location, expressed as latitude and longitude, of the observer of a given target is abbreviated as (lat, long) or  $(\phi_o, \lambda_o)$ . The geographic



**Figure 2.3:** The celestial sphere, viewed looking down at the Earth’s North Pole, showing the Vernal Equinox ( $\Upsilon$ ), with illustrative object *right ascension* ( $\alpha$ ) and *observer longitude* ( $\lambda_o$ ), demonstrating the concepts of hour angle ( $h$ ), Greenwich hour angle (GHA), and local sidereal time ( $\theta_L$ ).

location is normally also coupled with an elevation—the altitude above mean sea level. Lat:  $-90^\circ$  to  $90^\circ$ ; Long:  $0^\circ$  to  $360^\circ$ .

### Horizontal coordinates

The Equatorial (RA, Dec) coordinates of the target object, converted into a viewing angle above the horizon, the altitude or elevation, and the direction as a horizontal angle from North, the azimuth. Azimuth and elevation/altitude values are abbreviated as (Az, El) or ( $A$ ,  $a$ ) and are derived from the viewer’s geographical location and the celestial coordinate of the target. The valid range for this parameter is Az:  $0^\circ$  to  $360^\circ$ , El:  $0^\circ$  to  $90^\circ$ . While the azimuth definition may appear problematic at the poles, where every direction is either South, or North—depending on the pole in question—the Greenwich meridian still applies and can be used as reference.

### Local horizon/Horizon mask

The local horizon around the telescope may be affected by the telescope position in relation to other structures, as well as surrounding topography. To avoid occlusion by nearby structures, or pointing at high terrain, a horizon mask may be constructed. This allows software to determine whether a given target would potentially be visible. The horizon mask can be represented as an ordered list of minimum pointing angles over an azimuth range of possibly several degrees.

### **Day/Night/Twilight**

The duration of day and night for any given location on the surface of the Earth, has a seasonal dependence; the further the location is from the Equator, the higher the seasonal variance. Similarly, the duration of twilight also depends on the season and geographical location. Generally the mean length of the night is 12 hours. Of course at the poles, the night or day duration may be several months.

## **Astronomical variables**

### **Celestial/Equatorial coordinates**

The position of the target object in the sky; the celestial coordinates of the object expressed in Right Ascension and Declination. It is abbreviated as (RA, Dec) or  $(\alpha, \delta)$ . Valid values for RA are  $0^{\text{h}}$  to  $24^{\text{h}}$ , and Dec are  $-90^{\circ}$  to  $90^{\circ}$ . Alternatively the RA parameter can be expressed relative to an observer's location, as the Hour Angle (HA) or Local Hour Angle (LHA).

### **Moon/Phase**

The light reflected off the Moon's surface is a major contributor to sky background illumination in optical astronomy. The phase of the Moon dictates the extent of sky brightening.

**Elongation/Separation angle**

In optical astronomy the position of the Moon in the sky relative to a target object determines the measurement precision of the light intensity from that object. Thus, a larger target-lunar elongation makes for better measurement precision.

**Conjunction/Occultation by the Moon and Solar System bodies**

The target object may be of a nature that you can observe the object during full moon, however, the Moon may be occluding the object. Unless this is the observation objective, this must be disallowed. Similarly so for all stars that lie in a narrow region about the ecliptic<sup>4</sup>. Most of the solar system bodies such as planets, and their moons, orbit about the Sun in this plane and targets in this region need to be evaluated for possible occultation by a solar system object.

**Object brightness**

Astronomical detectors can either exhibit non-linear responses or saturate when exposed to high levels of illumination. This places a limit on the maximum allowable brightness of a given target object. The converse is also true: an object may be too faint to discern with the limitations of a given detector.

The brightness of astronomical objects is specified using the magnitude scale, a logarithmic scale devised by Greek astronomer Hipparchus approximately 2000 years ago. It is an inverse scale where fainter stars have higher magnitude values. To give an indication of the scale, the brightest star in the night sky, Sirius, has a magnitude of  $-1.4$ , while the typical limit of the naked eye under very dark conditions is 6th magnitude.

---

<sup>4</sup>The ecliptic is the apparent path the Sun traces across the sky in one year, with respect to the background stars.

## Atmospheric conditions

### Extinction/Airmass

The longer the column of air that a ray of light passes through, the more the intensity of the detectable light decreases, due to scattering and absorption. Cumulatively, these losses are termed extinction. For particular sources, and particular applications, light losses due to the atmosphere may be of critical importance. This atmospheric extinction is parameterised in terms of *airmass*; extinction is normally expressed as magnitudes per airmass. At mean sea level airmass can range from 1, directly overhead, up to approximately 40 at the horizon. As the airmass is nominally measured from mean sea level, the elevation of the observer can be incorporated for more accurate values, to give a minimum value between 0 and 1. For a fixed observing location the airmass curve is fixed, and a relation between angle and airmass need only be calculated once.

### Seeing

As light passes through air, it may be diffracted and scattered due to turbulence and localised temperature differences. This manifests as twinkling, and is called *seeing* in astronomical parlance. Seeing is the diameter of the fuzzy blob that results from a long exposure of a point-like source through the atmosphere. This is measured in arcseconds and normally ranges upwards from about  $0''.4$ ; this low value is only really attainable during the best conditions at high-altitude telescopes on small islands. Since seeing is directly influenced by the immediate atmospheric conditions, and consequently accompanying or preceding weather conditions, it is one of the main dynamic scheduling parameters. This parameter must be continuously tracked during the nightly observations to ensure upcoming queued observations are well suited for the prevailing conditions.

### Weather conditions

Weather conditions include humidity, high wind speeds, fog, cloud cover,

precipitation, etc. Terrestrial telescopes are at the mercy of local weather conditions; optical telescopes especially so. The telescope may comprise sensitive electronic equipment and fragile optics; both of which are highly susceptible to moisture. High wind speeds may pose a mechanical or dust hazard to exposed instruments; the design of any equipment in operational setup commonly place an upper limit on the allowable maximum wind speed during observing operations. Monitoring of the weather conditions, and suspending operations when the prevailing conditions become too severe are normally handled by the telescope control system and are outside the control of the scheduler.

## Observation-specific variables

### Observation

The actual unit classified as the observation can consist of a single astronomy target pointing or a group of pointings and comparator targets or calibrator targets. If the group is small and a closely co-located the target may be defined as the evaluation target (see below). For larger groups of longer observations, the evaluation target is defined by the science proposal.

### Target

A target is specified using a coordinate pair, typically specified using the equatorial coordinate system, that can be used during constraint evaluation. It can be an actual astronomy source, or an observation reference position representing a group of targets. For example, the evaluation criterion for a group of targets may be: *a*) the rise time of the earliest target if this specific target's observation time is long enough to ensure that other targets in the observation have sufficient time to rise; *b*) for multiple targets or a series of linked observations that must be completed once the first observation has started, the equatorial coordinates of the reference sky position should be chosen as the centroid of all linked observations; *c*) or the rise time of the latest rising target—typically used when the on-target time is very short and

consequently the other targets must all be available before the observation can begin.

### **Target visibility**

The visibility of observation targets is a group constraint providing the aggregate result from a set of individual constraints. A target is defined to be visible if it is above the observation horizon and unobstructed at the time of evaluation and will remain above the observation horizon and unobstructed for the duration of the observation.

Some of the hidden complexities are to ensure the constraints take into account Eastern (rising) and Western (setting) targets, as well as far Northern (or Southern) targets that are only visible for some part of the observation term. Additionally, ensuring that any given target will not be obscured by an eclipse event during an observation—except in the case where this is the observation objective—which may be accomplished as a hard constraint specifying a separation angle or a softer constraint set by a percentage of time lost during observation—with all of the above dependent on the scientific goal and type and quality of observation required.

### **Observation term**

A quarterly or semester scheduling cycle is often specified by the observatory policy and associated to calls for proposals. These depend on the telescope and can be anything from quarterly to yearly.

### **Observation period**

The total time per day/night available for scientific observations depends on the instrument, the science and the time of year. For optical telescopes this may be the total time observations can be performed, specifically from astronomical evening twilight until astronomical morning twilight.

### **Visibility period**

The time period a target is visible depends on the instrument and the science. For optical telescopes this may be the total time observations can be

performed, specifically from astronomical evening twilight until astronomical morning twilight, whilst taking into account the rise- and set times of the object. Alternatively, for telescopes not limited to specified time, the observation period translates into the total time the target is visible.

### **Data acquisition time**

The time spent acquiring data refers to the time on target gathering observational data.

## **Operational considerations**

### **Scientific priority**

Proposals are assessed by the Time Allocation Committee (TAC) on the basis of technical feasibility and potential scientific impact. The policies and aims of a particular institution may also favour specific scientific goals above others. This leads to the assignment of priorities to observation programmes by the TAC. These priorities may be assigned to a project as a whole, or individual programmes in the project may be assigned different priorities.

### **Weights**

When prioritising the candidate target objects for observation during a given interval, say, on a given night, specific targets may be assigned additional weights according to some policy, or observational criteria. For instance, the aim may be to complete as many proposals as possible, so proposals with fewer remaining observations are given a higher weighting over proposals that have, for instance, 50 per cent completion. Conversely, the policy may be to do as wide a spread of science as possible, so proposals with few remaining observations may linger longer. In general, incomplete projects may not produce any scientific return and observations with a significant level of completion will be promoted.

Partners own a certain percentage of the telescope time, but usually not a specifically determined fraction on any given night—there needs to be some method to ensure fairness over a given time period, for example weekly,

monthly, quarterly, or per observation term.

### **Programme tags**

A given proposal may be put on hold due to some external factor. For example, the proposal may be part of a pilot study, and assessment of the already observed data needs to be completed in order to show that the study has merit. To allow these types of scenarios, additional flags may be added to a proposal. These could include information such as *active*, *complete*, or *suspended*.

### **TLE**

Two-Line Element (TLE) files are used to describe orbital elements (i.e. the ephemeris) of man-made space objects. Usually these values are subject to perturbations and must be updated regularly.

### **Wavelength**

Apart from optical wavelengths, other frequency bands can also be observed by radio/gamma/etc. telescopes. These instruments may be wideband instruments and a selected frequency range must be specified.

### **Filter wheel sequence**

Filters work by allowing light of a specific wavelength range to reach the detector while blocking other wavelengths. This increases the signal-to-noise ratio of the wavelengths of interest. Filter wheels are used to position a selection of filters in the optical path. Depending on the speed and directional movement capability of the wheel, the mount sequence of the filters in the wheel should be taken into account to minimise scheduling observation series that may each require filters located far apart in the filter wheel sequence.

### **Instrument/observation mode**

Telescopes are often fitted with a range of instruments; some of these offering different selectable modes. Depending on the implementation, the instruments may need to be fitted manually and may only be available for a certain time during a scheduling period.

**Zones of avoidance**

Mechanical obstructions within the dome, or line-of-sight obstructions outside the dome must also be taken into account when evaluating obscuration or pointing positions. Since these are generally known and static, they may be hard coded into the TCS, but may also vary depending on science target. Consider the example when the ACT points North at a high zenith distance while the dome is oriented to the South. Van Heerden (2011) notes that in this particular configuration the telescope may collide against the dome and get stuck. Even though this should not occur during normal operations, this particular configuration should be avoided when determining trajectories for telescope and dome movement.

**Telescope specific/Instrumental limits****Minimum pointing altitude/Minimum altitude**

The telescope may have specific mechanical constraints. It may not be possible, for example, to point below a given altitude angle, or tracking may degrade above a maximum altitude. The minimum pointing altitude may be a fixed number, or be an ordered list, equivalent in form to the horizon mask.

**Telescope pointing singularity**

The telescope may have some specific mechanical constraints which are inherent in the design. With an alt-az mount telescope, for example, pointing to, or tracking through the zenith is problematic. The pointing singularity may, of course, lie outside of the nominal operational envelope of the telescope. The telescope pointing singularity can be expressed as a list of no-go coordinates in a particular reference frame, or as a list of no-go coordinates with associated minimum avoidance angles.

**Target acquisition time**

This is the time the telescope takes to move from a particular on-sky position to point to a different position. The target acquisition time has several

components:

**Telescope slew time** The movement profile of the telescope can be very well characterised, so the target acquisition time can be estimated accurately. Also, telescope movements may not be commutable, thus the telescope slew path may also play a role.

**Dome rotation rate** Dome rotation is usually slaved to telescope position, so for most slews, the dome *keeps up* with the telescope. The dome may rotate at a different speed to that at which the telescope slews. Depending on the relative differences in the motion rates, the optimisation of target location selection may have to take these motion rates into account.

**Fine pointing** Fine pointing refers to the, possibly iterative, process of repositioning a telescope after a coarse pointing movement so that the target is within a given region in the telescope's field of view. This pointing adjustment is typically in the order of one to several arcseconds.

**Focusing** Once the target is acquired and centred, some additional focusing functionality—depending on the instrument—may be required.

**Dead/setup time** Some little additional time that may be needed to set apertures, move filters into position, focus, etc.

#### **Data readout/transport time**

In the event of low bandwidth, the time needed to write data to persistent storage may be of significance. The determining factor could also be readout time, but this is unlikely for small telescopes.

### **Mechanical parameters**

#### **Non-operational time/Procedural-time**

This can be used to identify time periods that the telescope is not busy performing scientific observations. This includes any start-up procedures at the beginning of a night, the shutdown procedures at the end of a night, and any system calibration observations that need to be performed periodically.

Ideally the calibrations should be scheduled to occur during a time that the telescope cannot perform scientific observations, for example during day-time or twilight for an optical telescope. However, this is not achievable under all circumstances, especially so when a specific calibration needs to be performed more frequently to ensure data integrity or data calibration, or the specific calibration is only possible during night-time. Also, there may be a need for on-sky calibrations, such as observing photometric standard stars or capturing flat fields.

**Scheduled downtime**

This can be planned maintenance periods or service cycles.

**Failure**

Mechanical-, electrical-, software-, or IT-related failures are unplanned, as a rule, and can therefore happen at any time. From a scheduling perspective, this is an interruption event and must be handled as it occurs.

## Chapter 3

# Automated scheduler

Conceptually, a scheduler must take as input a set of observations that has been requested, as well as the constraints peculiar to the observations and specific of the instrument/environment (Frank, 2000). The output will be some criteria derived from the optimisation of specified goals. Some easy and fairly reliable methods to achieve this are described by Granzer (2004) and Frank and Kürklü (2003). This chapter will discuss the theoretical detail behind implementing these methods in the strawman scheduler.

As described in Chapter 1, there are three criteria for a *good schedule*: *a*) fairness, *b*) efficiency, and *c*) sensibility. A fair schedule balances time allocations between users such that they all share good and bad observing times equitably. An efficient schedule is one that maximises instrument utilisation and strives to match observations with required conditions. A sensible schedule is one that attempts only those observations that are possible under the current observing conditions (Denny, 2004).

These requirements for a good schedule are translated into observational constraints that can be evaluated during scheduling (Granzer, 2004). Evaluation uses some optimisation of an objective function representing a per observation rank calculation based on the constraints (Frank and Kürklü, 2003). It should be noted that an observation specifies both hard constraints and soft preferences. The scheduling problem is to synthesise

a schedule that satisfies all hard constraints and achieves a good score according to an objective function based on the soft preferences (Swanson, Bresina, and Drummond, 1994).

## 3.1 Basics of scheduling

### 3.1.1 Planning

The scheduling of astronomical observations is typically conducted on several different time scales. Longer term planning deals with scheduling over the observation term given the approved science projects. The main aim of this type of planning is the equitable distribution of time among users/partners, as well as maximising scientific return. This phase only takes into account observational constraints that do not change, are known, or can be predicted/calculated very accurately.

Longer term planning deals with scheduling over the observation term, given the approved science projects. The main aim of this section is the fair distribution of time among users/partners, as well as maximising scientific return.

Optimisation for long-term planning is mainly driven by the aims of the observatory and is restricted by the constraints of the telescope. Observations can, and usually do, conflict. Longer term plans allow for better resolution of these conflicts to achieve optimal scientific output.

For several reasons, it may not be possible to execute all approved projects within a given time frame. Oversubscription is therefore permitted for the full cycle in order to ensure the complete use of available time.

Intermittent re-planning allows for the re-evaluation of observatory performance, which in turn allows for the re-evaluation of parameters or change of optimisation function. Furthermore, over the lifetime of the observatory, other constraints may be required such as those imposed by a new instrumentation or by change in operations.

The optimisation strategy for planning ignores dynamic conditions and assumes *a*) the problem-free execution of each observation, *b*) perfect knowledge of the time duration needed for each observation, and *c*) perfect fore-knowledge of the weather throughout the night (Denny, 2004). Given the complexity and size of the search space for long-term scheduling, it is obvious that this type of scheduling cannot be done in real time; therefore the focus does not have to be on time-efficient algorithms.

### 3.1.2 Scheduling

Following the broader planning phase, the scheduling phase is more focused on optimising the use of the telescope, minimising overhead and maximising science output.

Setting up a dynamic queue of observations available for execution, based on a subset from the planning section, allows the scheduler to focus on efficient use of telescope time and instrumentation setup. While planning decreases the number of observations to consider based on best-choice and other fixed constraints, setting up a selection of viable observations is subject to a large number of complex, heterogeneous constraints over both continuous and discrete variables. Even relatively simple schedules have to deal with geometric constraints, precedence constraints, mutual exclusion constraints and temporal constraints, all in the same problem (Frank, 2000).

Non-scorable observations, such as TOOs, are subject to their own unique scheduling rules, where there is nothing to optimise. The only goal is to ensure every required observation actually gets on the schedule given its individual constraints. Optimisation of other observations happens around these observations and will generally result in a less-optimal solution.

Some observations are naturally more interesting to the science community than others. However, due to the limited observation time, it may be necessary to observe a target many times, and so it may be more important to finish a sequence of observations on a given target rather than to start a new observation of another target. In order to ensure maximum viable science output for publication an observation rule may state

that once an observation is started it must be completed ahead of other observations still waiting to start, irrespective of rank (Frank, 2000).

### 3.1.3 Observations

The scheduler thus generates a queue of observations available to be executed based on predicted values. These values are generally allowed to be oversubscribed with the system continually processing the short-term viability of queued observations during observation runs, based on constant updates that can include additional observations or additional constraints or triggered observations.

Executing observations is extremely time constrained and minimal optimisation should be done. The emphasis is to ensure a balance between efficiency and sensibility.

Additional constraints may also come in the form of scheduling rules, which may in turn affect observation requirements. An example is linked observations: once the first observation in a linked set is scheduled, the rest must be scheduled without optimisation of the individual observations if the entire set is to be completed in one round.

This requires on-demand scheduling strategies, where the scheduler dynamically makes a *best* choice for the next observation, maximising science efficiency by executing the programmes with highest scientific value first and under the required observing conditions. In addition, the scientific use of telescope time must be maximised by having appropriate programmes ready for execution under a broad range of observing conditions, thus being able to adapt to changing conditions, new requests, and acquisition errors, while still maintaining reasonable efficiency (Denny, 2004).

## 3.2 The dispatch scheduler

Operational parameters may be general to astronomy or unique to a telescope. Other influences will depend on observatory policies and procedures such as those related to

long-term projects, or compensations for time loss due to TOO observations or similar programs.

In any queue scheduling methodology, the proper treatment of constraints on the observation is of paramount importance. Some of these constraints are explicitly given by astronomers, while others are implicit, due to the nature of instrument/telescope (Frank, 2000).

In order to decide which observation,  $n$ , to carry out, a per observation objective function is evaluated (Steele and Carter, 1997):

$$R(n) = f(n) \cdot \prod_{x=1}^{x=X} v_x(n) \cdot \frac{\sum_{m=1}^{m=M} \varepsilon_m(n)}{M} \quad (3.1)$$

For any observation constrained by  $X$  hard limits and  $M$  soft preferences:  $f(n)$  is a measure of fairness,  $\varepsilon_m(n)$  measures of efficiency and  $v_x(n)$  Boolean veto functions as measures of sensibility.

Constraints are normalised to ensure an equal impact on the calculation from all, and to prevent a situation where high-valued constraints have a high impact, while low-valued, high-importance constraints have no real effect on the rank calculation. Also, not all projects will have the same number of constraints and this must not unfairly bias some projects. The only influence on selection must be scientific relevance (Maartens, Martinez, and Van Rooyen, 2017).

Observatory time must be shared equitably between projects. The fairness function evaluates how equitable it is to perform a particular observation, based on the project's time allocation. The time allocated to partners is thus a form of observatory accounting and when this drops below a partner's share of time, the system must give higher preference to that partner (Kubánek, 2008).

The veto function has to prevent observations being carried out that are not possible

at the current time due to a number of Boolean constraints.

$$\prod_{x=1}^{x=X} v_x(n) = v_1(n) \cdot v_2(n) \cdot \dots \cdot v_X(n), \quad (3.2)$$

where  $v_x(n)$  describes the constraint limits.

The purpose of the efficiency merits is to decide which observation to carry out, at any given moment in time, considering observatory policy, scientific importance and observing conditions (Steele and Carter, 1997).

$$\sum_{m=1}^{m=M} \varepsilon_m(n) = \beta_1 \varepsilon_1(n) + \beta_2 \varepsilon_2(n) + \dots + \beta_M \varepsilon_M(n), \quad (3.3)$$

where  $\varepsilon_m(n)$  describes the constraint equations, each with an optional weighting factor  $\beta_m$ .

### 3.2.1 Astronomical veto functions

Astronomical constraints that can be considered as *hard* constraints, are generally related to observational limits.

Positional fitness depends on the target position relative to some time standard and the observatory location. One of the most obvious position conditions is target visibility. In order to observe a target, it must be in a part of the sky visible to the telescope at some time during the observation period.

In terms of actual target sky visibility, the current definition will consider a target *visible* if the target elevation is above the telescope's local horizon during the observation period.

$$v(\text{visible}) = 1 \forall \theta_{\text{target}} > \theta_{\text{horizon}} \in (N_{\text{start}}, N_{\text{end}}) \quad (3.4)$$

---

Function 3.1: Veto(visible)

---

 $N_{start} \leftarrow$  observation period start time

 $N_{end} \leftarrow$  observation period end time

**if**  $N_{start} \leq target.minangle\_time$ 

    **and**  $target.minangle\_time \leq N_{end}$ 

    **then**

        |  $v(visible) \leftarrow$  **Permit**

    **else**

        |  $v(visible) \leftarrow$  **Prohibit**


---

Target brightness evaluation is based on the instrument sensitivity limits related to the source target properties. The brightness of the object must be low enough not to saturate the instrument, but high enough to provide a viable observation.

$$\begin{aligned}
 & \text{instrument noise limit} \leq \text{Target brightness} \\
 & < \text{instrument brightness limit} \qquad (3.5) \\
 & v(magnitude) = 1 \in [\text{noise limit}, \text{brightness limit}]
 \end{aligned}$$

---

Function 3.2: Veto(magnitude)

---

**if**  $instrument\ brightness\ limit \leq target\ brightness$ 

    **or**  $target\ brightness < instrument\ noise\ limit$ 

    **then**

        |  $v(magnitude) \leftarrow$  **Prohibit**

    **else**

        |  $v(magnitude) \leftarrow$  **Permit**


---

Lunar phase and elevation not only influences sky brightness calculations, but also relates as a hard limit to observational *brightness* conditions and can be defined in

terms the percentage of the visible surface disc that is illuminated (PLI).

$$\text{Lunar brightness} = \begin{cases} \text{dark, if PLI} < 0.4 \\ \text{grey, if PLI} < 0.7 \\ \text{no constraint, True} \end{cases} \quad (3.6)$$

$$v(\text{dark}) = 1, \text{ if dark}$$

$$v(\text{grey}) = 1, \text{ if dark} \cup \text{grey}$$

$$v(\text{any}) = 1, \text{ if dark}^c \cap \text{grey}^c$$

Note that the PLI values above are not mutually exclusive; targets that permit *grey* time may also be scheduled during *dark* time, for instance, in the event that no targets with a *dark* requirement are available.

---

Function 3.3: Veto(sky brightness)

---

**if** dark PLI < *moon phase* **then**

    |  $v(\text{dark}) \leftarrow \mathbf{Prohibit}$

**else**

    |  $v(\text{dark}) \leftarrow \mathbf{Permit}$

**if** grey PLI < *moon phase* **then**

    |  $v(\text{grey}) \leftarrow \mathbf{Prohibit}$

**else**

    |  $v(\text{grey}) \leftarrow \mathbf{Permit}$

---

Conditions are considered to be photometric if the seeing is better than 1.3 arcseconds.

$$\text{Seeing} = \begin{cases} \text{poor, if seeing} \geq 1''.3 \\ \text{average, } 0''.7 < \text{seeing} < 1''.3 \\ \text{good, if seeing} \leq 0''.7 \end{cases} \quad (3.7)$$

### 3.2.2 Astronomical efficiency functions

General constraints are most important during the optimisation of the observation scheduling. Since the *strictness* of these soft preferences depends on the observation, soft constraints are defined using merit functions which can be adjusted to make the constraint more, or less, stringent.

#### (a) *Airmass merit*

The closer a target is to the horizon, the more atmosphere the signal must pass through. Atmospheric absorption at lower elevations may cause degraded results. The general preference is to observe targets at as high elevation as possible. Airmass can be used to assign lower weights as the targets get closer to the horizon, thereby favouring observations at higher elevation.

$$\varepsilon_h(\text{airmass}) = \frac{1}{z(h)^\alpha} \quad (3.8)$$

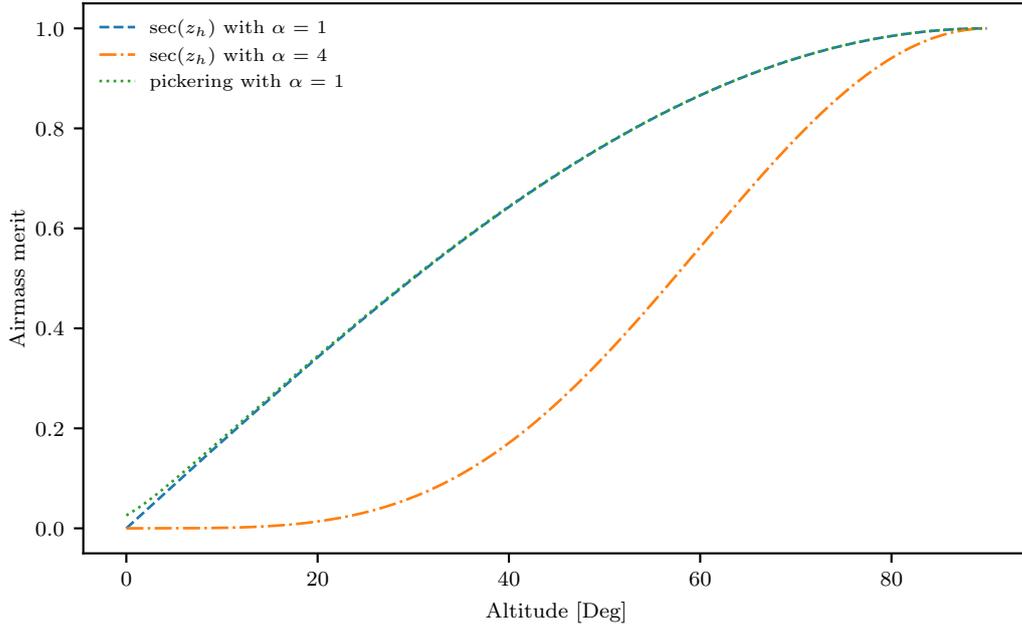
for the airmass at the observation reference position using the  $\alpha$  coefficient to define the steepness of the merit (Figure 3.1).

#### (b) *Separation angle merit*

The target must not at any stage of an observation approach within a specific minimum angular distance from the Moon. Separation angles may be dependent on the observation wavelength with different criteria between longer and shorter wavelengths, or brightness of target and comparator pair (Figure 3.2). It is advised that the separation angle be chosen as narrow as possible since very strict phase and angle requirements may drastically reduce the time period in which the observation can be carried out, and hence a lower probability that it would be successfully completed.

$$\varepsilon(\text{separation}) = \left( \frac{\theta(\text{target}, \text{Moon}) - a}{b} \right)^c \quad (3.9)$$

For a separation distance  $\theta$  and separation limit  $a$ , parameters  $b$  and  $c$  are used to shape



**Figure 3.1:** Possible airmass merits depending on airmass model selection. The standard homogenous plane-parallel atmosphere approximation,  $\sec(z_h)$ , compared to the Pickering model—the Pickering model (Pickering, 2002) is currently the best model for high accuracy near the horizon. Both models scale the strictness of the airmass merit using the steepness parameter  $\alpha$ .

the strictness of the merit.

Equation 3.9 illustrates the separation merit for the Moon. Similar merits can be defined for other solar system bodies.

(c) *Target altitude merit*

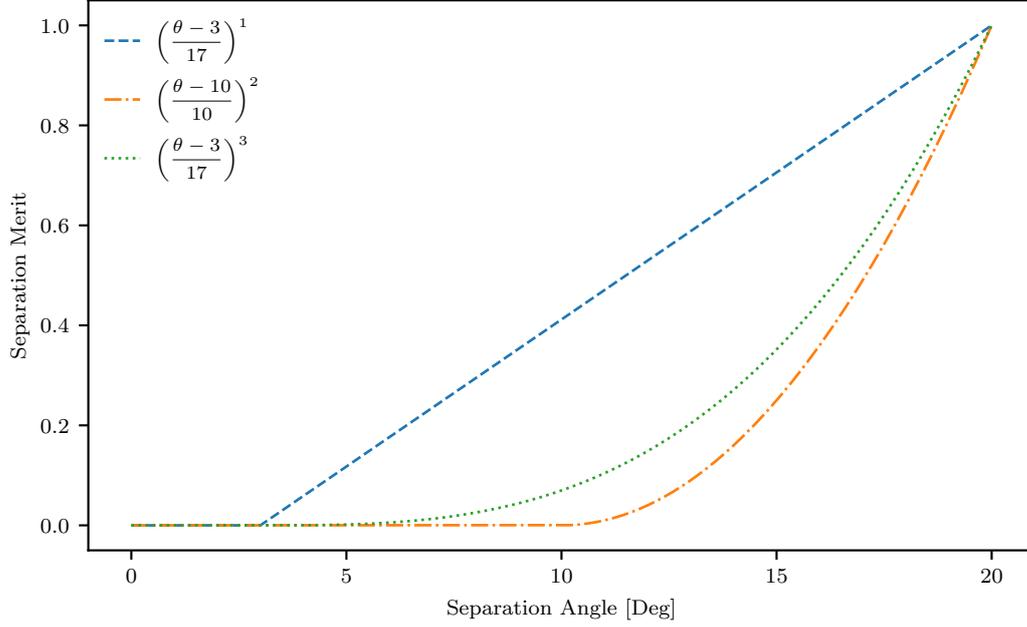
When atmospheric effects are less important, but observations at higher altitude is still preferred due to mechanical or structural considerations; a simple piece-wise linear relation based on the altitude of the observation reference position can be used.

Equation 3.10 provides such a calculation:

$$\varepsilon_a(\text{altitude}) = \frac{a - \max\{E_{\min}(\text{horizon}), E_{\min}(\alpha_{\text{target}})\}}{\min\{E_{\max}(\text{limits})\} - \max\{E_{\min}(\text{horizon}), E_{\min}(\alpha_{\text{target}})\}}, \quad (3.10)$$

and is graphically illustrated in Figure 3.3.

Parameters used in Equation 3.10 are defined as  $a$ , the current altitude of the first target



**Figure 3.2:** Separation angles merit allows for softer limits as targets approach solar system objects, thus improving the probability of observation.

of the candidate observation, derived from the centroid over all linked observations in a linked sequence;

$E_{min}(horizon) = w(a)$  for some observatory related horizon mask,  $w$ ;

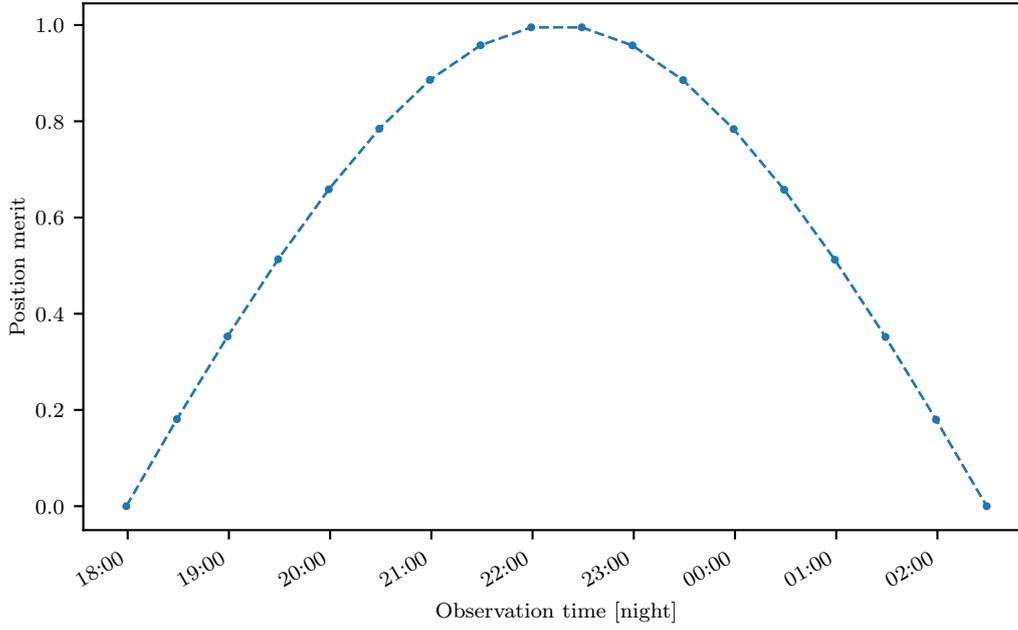
$E_{min}(\alpha_{target})$  target visibility limit;

$E_{max}(limits) < zenith\ limit$  with  $zenith\ limit$  a singularity for alt/az mounts.

For a southern hemisphere observatory at latitude,  $\phi_o$  and an object with declination  $\delta$ , the minimum and maximum altitudes are calculated:

$$E_{min}(horizon) = \begin{cases} -90^\circ - (\phi_o - \delta), & \text{if transit during observation period} \\ \min\{E(N_{start}), E(N_{end})\}, & \text{otherwise} \end{cases},$$

$$E_{max}(horizon) = \begin{cases} 90^\circ + (\phi_o - \delta), & \text{if transit during observation period} \\ \max\{E(N_{start}), E(N_{end})\}, & \text{otherwise} \end{cases},$$



**Figure 3.3:** Simple position evaluation, piece-wise linear calculation over 30 minute intervals. The merit favours higher sky location during positional evaluation.

where  $N_{start}$  ( $N_{end}$ ) defines the start (end) of the observation period.

---

Function 3.4: Efficiency(target altitude)

---

$$\begin{aligned}
 \text{elevation}_{min} &\leftarrow [ \\
 &\quad \min(\text{observer horizon}(N_{start}), \text{observer horizon}(N_{end})), \\
 &\quad \text{instrument minimum altitude}, \\
 &\quad \text{horizon mask}(\text{target azimuth}) \\
 &] \\
 \text{elevation}_{max} &\leftarrow [ \\
 &\quad \text{instrument zenith pointing limit}, \\
 &\quad \text{altitude}(\text{target transit}) \\
 &] \\
 \varepsilon(\text{target altitude}) &\leftarrow (\text{target altitude} - \max(\text{elevation}_{min})) \\
 &\quad / (\min(\text{elevation}_{max}) - \max(\text{elevation}_{min}))
 \end{aligned}$$

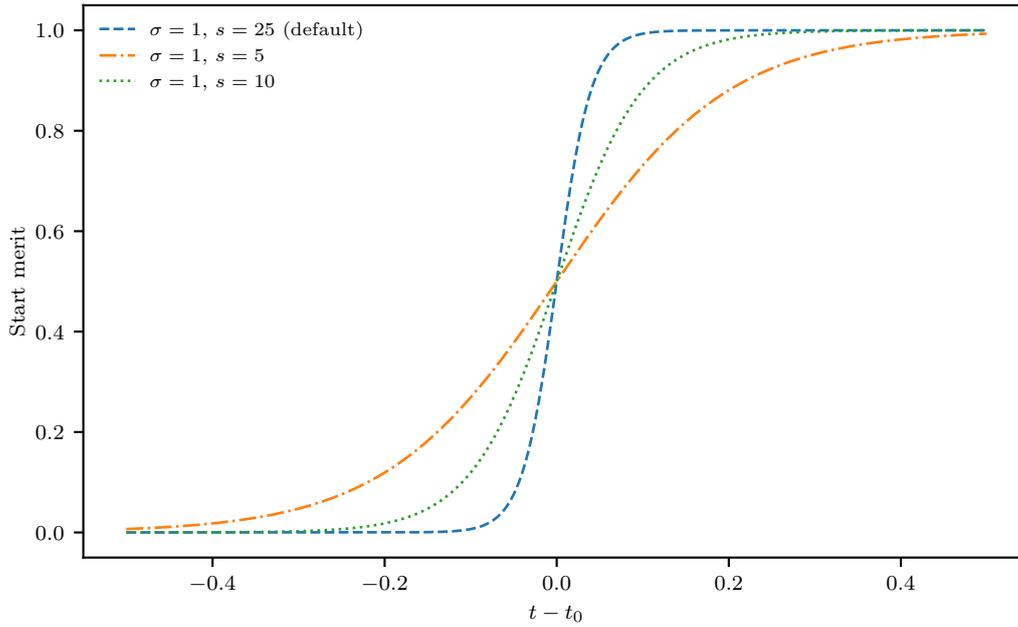

---

(d) *Rise and set time merit*

In addition to the target position, timing related constraints are also very important

for optimal scheduling.

As the night progresses and targets rise, these targets become part of the scheduler options and must be evaluated depending on the strictness of starting observation at around the rise time, Figure 3.4. While, for setting targets a preference may be given to favour the observations closer to termination—shown in Figure 3.5.



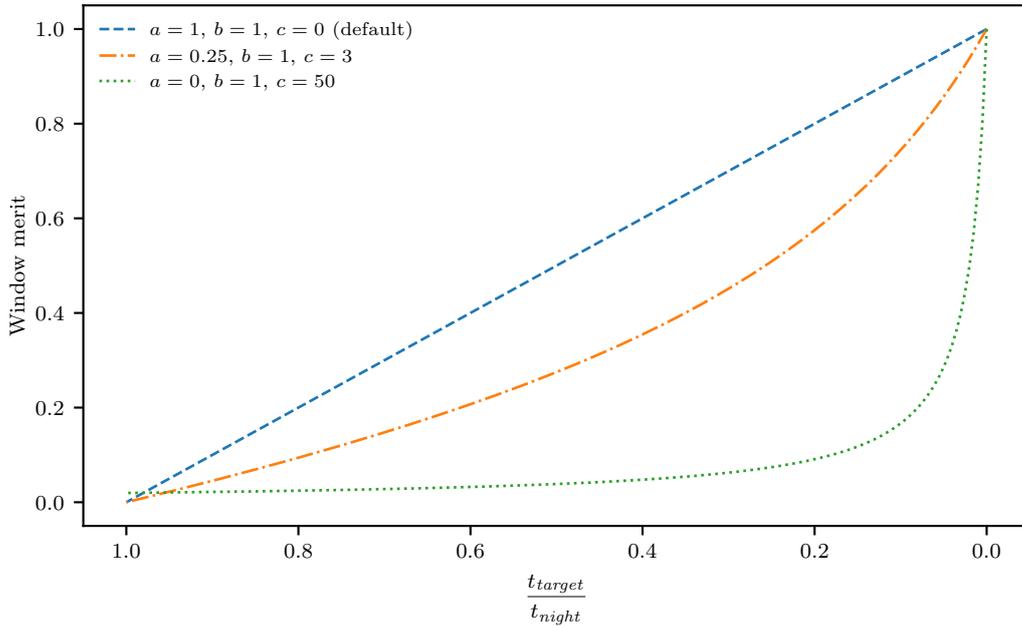
**Figure 3.4:** Merit indicating strictness to start observation around target rise time.

$$\varepsilon(\text{boundary}) = 0.5 \left[ 1 - \tanh \left( s \frac{t - t_0}{\sigma} \right) \right], \quad (3.11)$$

relaxed by the gradient  $\sigma$  as time approaches the termination boundary and  $t - t_0$ .

Together with evaluating setting targets, the window merit can also represent the evaluation of time remaining to complete observations for a given project. Figure 3.5 shows a window merit that increases the selection weight as the target observation window that can be used to evaluate both setting targets, as well as projects approaching completion, shortens (Granzer, 2004).

$$\varepsilon(\text{window}) = -a \times t_r + \left( \frac{b}{1 + c \times t_r} \right), \quad (3.12)$$



**Figure 3.5:** A merit that increases the selection weight as the time remaining to observe the target in the current night decreases.

where  $t_r = \frac{\Delta t_{target}}{\Delta t_{visible}}$  is the ratio of the target observation window over observation time remaining. The parameters  $a, b, c$  in this merit are only used to control the steepness of the rise.

### 3.3 Putting it all together

After identifying relevant parameters and describing their relation to the observation using the hard limit veto functions, as well as setting optimisation evaluation using the merit functions, an observation is ready to be scored using the objective function and can now be added to the scheduler.

For easy and continuous evaluation across all viable observations, the vetoes and merits must be structured in some logic algorithm; while, the data must be kept consistent with observations and accessible on demand. Also, a memory of evaluation must be kept and non-viable observations removed.

All of which brings together the equations of Chapter 3 into the implementation of

Chapter 4. The strawman scheduler implementation addresses the problem of combining the mathematical constraints to distribute observations over time on a single instrument. For the ACT this means selecting a sequence of observations starting at some time after sunset,  $N_{start}$ , and ending before sunrise,  $N_{end}$ .

## Chapter 4

# Implementation

The queue scheduling algorithm developed by the Stratospheric Observatory for Infrared Astronomy (SOFIA) (Frank and Kürklü, 2003) provides a well designed automated scheduling methodology. SOFIA, however, uses the dynamic constraint satisfaction problem (DCSP) optimisation strategy. This is a very good optimisation strategy, but is a greedy methodology that is computationally very time intensive. For the proof-of-concept strawman implementation the simpler rank function introduced by the *on-demand* scheduling strategy (Granzer, 2004) and discussed in Chapter 3, will be used as an alternative.

The basic queue scheduling problem can be stated as a permutation problem. Every ordered list,  $\{S(0), \dots, S(N_s - 1)\}$ , of observations represents a possible solution and the optimum solution amongst them is found using some evaluation method (Gómez de Castro and Yáñez, 2003). The goal of the strawman scheduler is to construct a good observation plan that can be queued and executed without much human interaction. Generating this full observation plan is very time consuming since the system needs to evaluate all observations in a set of available observations  $\mathcal{O}$ , find those that can feasibly be scheduled at a given time  $h$ , identify the best observation from that subset and add it to the observation plan  $\mathcal{P}$  being built. The time  $h$  is adjusted accordingly and the evaluation repeated as long as there are observations available to be queued. In order

to find the best queue sequence, the process is repeated for a number of permutations.

Note, in the algorithms below, the following notation applies:

- $\emptyset$  is the empty set, and
- $\parallel$  indicates concatenation. This has the effect of including the observation in an unordered set (e.g.  $O \parallel o$ ) or appending it to an observation plan, which is an ordered set (e.g.  $P \parallel o$ ), or setting the start time of the next observation to be appended to an observation plan (e.g.  $P \parallel h$ ).

## 4.1 Algorithm development

The strawman algorithms were developed from the SOFIA algorithms, but avoid the explicit use of flight-related parameters since these can be assigned more generally to extend the algorithms to ground based telescopes.

The fundamental algorithm of the SOFIA scheduler is the `ForwardPlan` algorithm (Frank and Kürklü, 2003). The `ForwardPlan` algorithm consists effectively of two sections: the first takes a list of possible take-off times and does a quick build of a short schedule for each start time. It uses these short queues to find the best time to start the observation flight. After this step completes, the queue construction part of the algorithm uses the chosen take-off time to build an optimal observation queue. These two sections of the algorithm are completely independent and have been split up into separate functions for the strawman scheduler.

`StartTime`, Algorithm 4.1, implements the initial section of the `ForwardPlan` algorithm. Although the SOFIA implementation of this is specifically to identify a take-off time, this concept is also valuable for under-subscribed telescopes. By being able to inspect a range of times to start observing, you can optimally distribute observations over the time available, instead of blindly scheduling targets as soon as they become visible, thus producing a sub-optimal schedule.

---

Algorithm 4.1: StartTime

---

**Input:** set of possible start times:  $H$   
 set of available observations:  $O$   
 look-ahead length:  $K$   
**Output:** start time of plan:  $h$

$P \leftarrow \emptyset$   
 $Q \leftarrow \emptyset$

**for** each start time  $h \in H$  **do**  
    $O' \leftarrow O$   
    $P' \leftarrow P \parallel h$   
    $P' \leftarrow \text{LookAhead}(P', O', K)$   
    $Q \leftarrow Q \cup (h, \text{Evaluate}(P'))$

**if**  $Q$  not empty **then**  
    $h \leftarrow \text{Select}(Q)$

**return**  $h$

---

The `StartTime` algorithm tries each observation that is feasible at the suggested start time,  $h$ , as the first observation and builds a short queue with length of the look-ahead length,  $K$ , thus obtaining a per start time set of possible short queues. The score from the highest scoring queue,  $P'$ , is selected and stored, along with the start time,  $h$ —which was used to generate this queue—in a score keeper list,  $Q$ . After scores have been obtained for each trial start time, the start time with the highest score is returned.

Building the optimised queue is done in `ForwardPlan`, Algorithm 4.2. This algorithm takes each available observation, in turn, and does an exhaustive search over the remaining candidate observations until there is nothing feasible left to schedule. The feasibility test also takes the remaining night length into account so that the algorithm terminates when no observations remain that fit into the remaining night length; that is to say, when the night has been fully scheduled.

In other words, starting at time  $h$ , returned from the `StartTime` algorithm; select the next available observation as a candidate observation,  $o$ ; do an exhaustive search over all remaining unscheduled observations to obtain a number of short queues of look-ahead length,  $K$ ; select the highest scoring of these queues,  $P'$ , and add this queue, along with the candidate observation  $o$ , which was used to generate it, to a set of candidate queues,  $Q$ . Repeat this process for all available observations to obtain a set

Algorithm 4.2: ForwardPlan

---

```

Input: start time:  $h$ 
         set of available observations:  $O$ 
         look-ahead length:  $K$ 
Output: observation plan:  $P$ 

 $P \leftarrow \emptyset$ 
 $P' \leftarrow P \parallel h$ 
while  $O$  not empty do
   $Q \leftarrow \emptyset$ 
   $P' \leftarrow \emptyset$ 
  for each observation  $o \in O - P$  do
    if Feasible( $o$ ) then
       $P' \leftarrow P \parallel o$ 
       $O' \leftarrow O \parallel o$ 
       $P' \leftarrow \text{LookAhead}(P', O', K)$ 
       $Q \leftarrow Q \cup (o, \text{Evaluate}(P'))$ 
    if  $Q$  not empty then
       $o' \leftarrow \text{Select}(Q)$ 
       $P \leftarrow P \parallel o'$ 
      remove  $o'$  from  $O$ 

return  $P$ 

```

---

of short queues and candidate observations. From this set select the highest scoring queue and add the candidate observation, used to generate this queue, to plan  $P$ . Thus, plan,  $P$ , is extended by repeating this next observation selection process until there are no more feasible observations left, thereby producing a highly optimised queue<sup>1</sup>.

The generation of the short queues is the workhorse of the queue generation process. The `LookAhead` algorithm is given in Algorithm 4.3. It functions as follows: until the observation queue,  $P$ , has been extended by the number of look-ahead steps,  $K$ , and while there are unscheduled observations available, take a feasible observation and extend the queue by this observation and evaluate the queue's score in order to decide which observation provides the best scoring queue. Do this for all feasible observations and select the observation that results in the highest queue score as the *next* observation,  $o'$ . The iterative cycle to select each next observation evaluates a selection over a number of steps into the future. In other words, each cycle generates a

<sup>1</sup>It should be noted that the optimised queue generation step either ignores dynamic observation conditions or implements models such as predicted weather patterns.

Algorithm 4.3: LookAhead

---

```

Input: observation plan: P
         set of available observations: O
         lookahead distance: K
Output: observation plan extended by K steps: P
repeat K times
  Q ← ∅
  for each observation o ∈ O − P do
    if Feasible(o) then
      Evaluate the rank function score of o
      Q ← Q ∪ (o, Score(o))
  if Q not empty then
    o' ← Select(Q)
    P ← P || o'
    remove o' from O
return P

```

---

throw-away schedule into the near future, for each candidate observation, to determine the best candidate to schedule by taking into account the possible observations that may follow it.

## 4.2 Database

The algorithms described in Section 4.1 form one of the pillars of the scheduler. The other pillar is the database on which the algorithms operate. The database contains all of the information on the targets, merits, and constraints. In this section we describe the development and structure of the database.

The reason for looking at the database design is to verify that it is possible to store the observation targets, constraints, and related data in a manner that allows for easy and quick retrieval. It is not hard to paint yourself into a corner with a restrictive design that does not leave options for future expansion, refinements and alterations. Thus it is important that the storage solution should be general enough, and without being restrictive, so that it can cater for usage patterns in the future that cannot be foreseen during design time. Note that this criterion holds specifically for the merit and veto

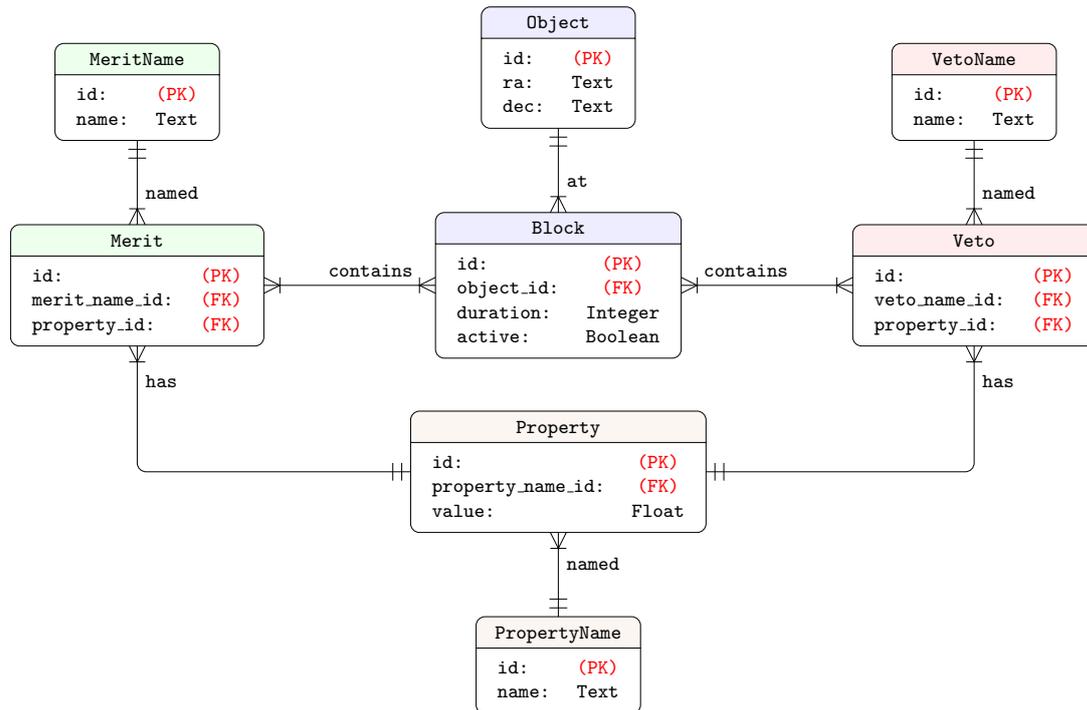
parameters, as not all possible merits and vetoes may be known at design time.

This section explores a possible extensible storage layout that will try to allow future additions and alterations, specifically with regards to veto and merit parameters. Of course this does not try to cater for the full database schema required for the general operation of an observatory, or a single telescope for that matter; it simply focuses on the data relevant to scheduling observations.

The initial implementation, shown in Figure 4.1, was structured with tables for: **target**, **block**, **merit**, **veto**, and **properties**. The **target** table holds the input parameters that define sky position as right ascension and declination. It must be noted that the assumption is all target equatorial coordinates are astrometric J2000 catalogue positions. To minimise duplication, identical targets are not repeated; rather, targets are associated to projects, with individual projects assigned a unique programme identifier once a proposal has been accepted. Following a modular design paradigm, the unique identifiers should be stored as part of the proposal management database which is not addressed in this development. To tie the proposal to the observations, defined by the **block** table, a 1-to-many (or possibly a many-to-many) linking table—not included in the prototype database schema—is required.

Possible future expansion is to include targets under human friendly target names. However, each catalogue defines a unique designation for each target object; designations for corresponding targets generally differ between catalogues. Current design strategies for time-domain astronomy is to develop an ecosystem of telescopes with a central hub, housing relevant catalogues and target information across telescopes, forming part of the Target and Observation Managers (TOMs) network (Street et al., 2018). Thus it would make sense to join this global effort and work to add a TOMs application programming interface (API) as extension to the scheduler database at a future date.

The **block** table defines an observation block and contains the observation parameters such as observation duration, priority, earliest start time and latest end time. Blocks



**Figure 4.1:** Initial database logical model in crow's foot notation.

are used as the building elements of a schedule.

The reason the `target` is kept separate from `blocks` is to prevent the duplication of data; this is a generally accepted design principle for relational database design. Since the same target may conceivably be associated with multiple observations, repeating the target coordinates for each observation introduces the possibility of errors. These errors may be caused when a particular value is initially entered incorrectly by the user, or by a target that may require adjustment, or refinement, at a later point in time. Such a modification would then require multiple updates in multiple places, any of which may introduce fresh errors of their own.

In turn, the `merit` and `veto` tables hold the function references and strictness parameters of the observation specific constraints. The `properties` table contains the values for each merit, or veto. For example, the elongation veto needs a minimum distance value, as well as the name of the celestial object this distance relates to.

The central table is the `block`, with a 1-to-many relationship to the `target` and many-

to-many relationships to the `merit` and `veto` tables. The `merit` table has a 1-to-many relationship with the `merit-name` table, to prevent duplication of merit name strings, since any duplication brings about the possibility of inconsistencies, in this case, of merit name spellings or capitalisation. A similar 1-to-many relationship is defined between the `veto` and `veto-name` tables.

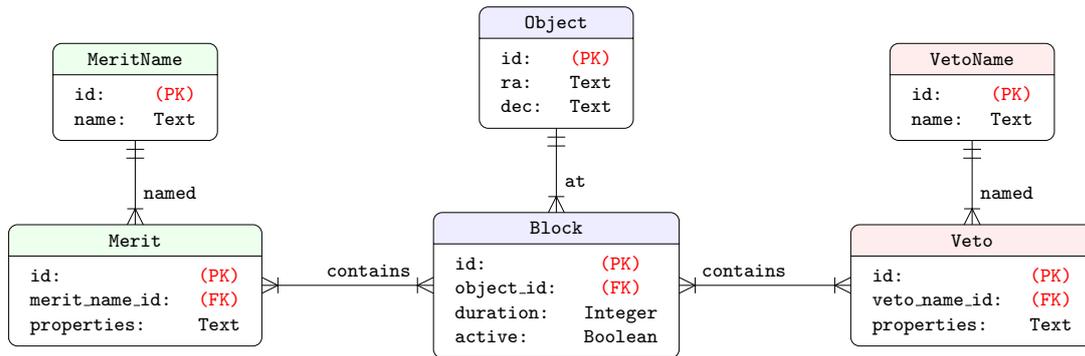
Both the `merit` and `veto` tables have a 1-to-many relationship to the `properties` table, which in turn has a 1-to-many relationship to the `property-name` table. Additionally, the `properties` table has a *value* text column to hold the string representation of the applicable property value—the storage type of this column was chosen to be as generic as possible, as some values may be integers, while other values may be floating point numbers, and yet other values may be strings.

This sharing of the `properties` table by the `merit` and `veto` tables is not ideal as a specific property carries no indication of whether it contains a merit or a veto value. An alternative to this is having a `merit-property` table and a `veto-property` table which have exactly the same structure; this may have been a better design decision, and may have been the next step in the iteration of this design, had it not been refined in a different way.

The use of a `properties` table had further disadvantages: parsing the property text strings proved hard. The idea of instantiating an object given a textual representation sounds easy, but it introduces all kinds of special case treatments, which is not ideal where the type of the object isn't known at design time. This means that all of the unique cases that might arise in the future cannot be catered for in a generic fashion.

The preceding discussion brings one in a roundabout fashion to think about storing metadata, or data about your data, in the database. Thus, you have tables describing the data that can be found in other tables. Generalising the idea produces a solution where data about other data is stored in the same table as the data, a so-called entity-attribute-value (EAV) model.

Gorman (2006) points out the pitfalls of the EAV model. As a comment to an online



**Figure 4.2:** Database logical model in crow's foot notation.

article, where Kyte (2009) explains why EAV model is not in widespread use, a reader asks about using this model to address their unique storage problem. Kyte responds that the data should rather be stored in the database as a blob of Extensible Markup Language (XML).

A refinement of this XML suggestion is to use JavaScript Object Notation (JSON) instead. JSON uses a rather less verbose notation, and while it is not as expressive as XML, what it allows one to represent is quite sufficient for the purpose at hand. Indeed, some database implementations handle data encoded as JSON natively, and can query the values directly from the JSON key-value store—it is thus not necessary to query a particular value from the database and parse it in the application; the database does this transparently. Therefore, the attributes can be stored as a JSON string in a column in both the `merit` and `veto` tables.

The final design is shown in Figure 4.2. This design has drawbacks though: the `properties` column of the `merit` and `veto` tables may contain duplicate data between different merits and/or vetoes. The design may be altered to have a `merit-property` and a `veto-property` table that contain the JSON data for the merits and vetoes; a future iteration of the database design might explore this option. A more serious drawback is that a particular `properties` entry may contain completely erroneous data that is not able to be parsed. One will only realise the latter when one queries the particular erroneous merit or veto row, and tries to parse the data. A way to guard against this possibility is to have a periodic process that selects each merit and veto individually,

and verifies that the JSON data is parseable. Another way is to have a post-commit trigger in the database, which fails the commit if the data one tries to commit to a particular merit or veto is invalid; this would prevent erroneous data from making in into the database in the first place. The strawman scheduler does not bother with this as the amount of data that is stored in the database is small, and the failure to parse data currently poses very little risk.

## Chapter 5

# Testing and verification

It is fairly obvious that computation time for the strawman scheduler is dependent on the number of observations that can make up a permutation. However, it has the advantage of always producing an optimal queue and it is easy to analyse the sensitivity of the scheduler to any of the constraints. Additionally, the scheduler introduces a fairness function to represent policy and procedure as well as scientific priority as part of the optimisation. This chapter describes some of the basic verification used to evaluate the implementation as per the scheduling design and optimisation strategy presented in the previous chapters.

Fundamental to the scheduler is the score function presented in Equation 3.1. The definition of this score function parametrically represents the three major aspects of the scheduling strategy, namely fairness, efficiency and sensibility. During the development of the strawman implementation, it is essential that verification tests be used throughout—not only to validate the correctness of the implementation, but also to ensure that the strategy of the score function is correctly captured in the Python implementation.

The strawman scheduler addresses the problem of distributing time on a single instrument. For an optical telescope, this means selecting a sequence of observations starting at some time after sunset,  $N_{start}$ , and ending before sunrise,  $N_{end}$ . This night length

restriction provides an obvious choice for a first verification step. The veto constraint validating target visibility determines whether an observation can be done, as well as the observation start time. This hard constraint must prevent an observation to start before sunset or end after sunrise, which is a fundamental restriction to the `StartTime` algorithm presented in Algorithm 4.1.

Secondly, the strawman also focuses on off-line searches to obtain the best feasible queued solutions, allowing the user to evaluate how these *optimal* solutions are associated with the observations and deals with observational and site-specific constraints. Where possible, evaluating individual merits will assist in verifying that the constraint is relevant to a typical observation. Evaluating the effect of updates/changes to the merit strictness could indicate how a given parameter could impact the observation evaluation and thus its scheduling times. By manipulating the expected impact on the observation schedule, the verification step tests the queue selection procedures implemented in the `ForwardPlan` and `LookAhead` algorithms of Algorithms 4.2 and 4.3.

Thirdly, the scientific priority of a program is a constant value that is assigned by the independent TAC process. As such it is not explicitly a merit function, but rather should be considered as an importance weight that should favour the more scientifically interesting observations. That said, the priority parameter can be implemented both as part of the optimisation sum of the score function, or, alternatively as in Equation 3.1, a definite weight affected only by time distribution fairness. Evaluating both implementations during verification will help to identify which of the two will be the preferred option for optimal astronomy scheduling.

Finally, combining merits and the priority measure will show that the policies and procedures of the observatory can be met.

It should be noted that results obtained from the verification tests are evaluated by visually inspecting graphs showing per observation scoring in relation to optimal schedule score achieved, as shown in Figure 5.1. Also assessed is the observation distribution over the available night time for the selected observation schedule, Figure 5.2.

## 5.1 Test targets

For the verification step the observation *targets* are very contrived but this allows one to build trust in the implementation by ensuring the correct handling of the basic functionalities highlighted above. Additionally, the selection of these targets makes visual verification, using displays of results over time, quick and easy.

The biggest issue is that the per observation evaluation, done by the scheduler, uses the target celestial coordinates to evaluate the scores over time and thus observation placement in the queue. Yet, programmatically manipulating the observations to evaluate their updated locations in a queue is easiest if it could be structured as a time manipulation related to the nightly time line.

As a result, the test target generator was created for verification and acts as a translator that takes time offsets as inputs and constructs celestial targets for the scheduler to use.

The test source generator requires three values to generate a target for scheduling:

- the target rise time with respect to sunset,
- the expected maximum elevation, and
- the observation duration.

Implicitly the function also uses values such as:

**the date** that the targets need to be valid for, in essence the time the resulting schedule should be prepared for, as well as

**the observer's location**, or in other words, the geographical position of the observer: latitude, longitude, and elevation.

These implicit values are needed to construct the fake *targets* that the scheduler will use to generate the observation queue. Specifying the observer's position, the date and time, and the maximum elevation of the target, provides enough information to generate a *celestial coordinate*.

The maximum elevation is the celestial object's elevation at meridian crossing. The object's declination is then either  $\delta = 90^\circ - (\phi_o + a)$ , for the case when  $a < 90 - \phi_o$ , or  $\delta = -90^\circ + (\phi_o - a)$  when  $a > 90 - \phi_o$ , where  $a$  is the object's culmination elevation,  $\delta$  is the declination, and  $\phi_o$  is the observer's latitude. In the trivial case where  $a = 90 - \phi_o$ ,  $\delta$  is of course  $0^\circ$ .

This is followed by calculating the compass direction the target will rise in using spherical trigonometry,  $A = \cos^{-1}(\sin(\delta)/\cos(\phi_o))$ . At sunset the altitude angle is  $0^\circ$ , giving an azimuth/altitude direction of  $(A, 0)$ , which can then be converted to a RA/Dec coordinate pair by using an astronomy software package such as Ephem<sup>1</sup>. Thus, by knowing when sunset is on the day in question, and adding or subtracting the rise time from that, the target's RA is obtained.

As an example, consider a target that rises half an hour before sunset on 16 March 2016, culminating at an altitude of  $65^\circ$ , with an observer at SAAO near the town of Sutherland. The calculated celestial coordinates, that is the RA  $\alpha$  and Dec  $\delta$ , of the target will be,  $(\alpha, \delta) = (14^{\text{h}}5^{\text{m}}7^{\text{s}}28, -7^\circ18'9''4)$ .

The reader should take note that the calculation used to generate the verification targets will be invalid when it is presented with a circumpolar coordinate, or in other words, a target that does not cross the horizon.

## 5.2 Basic functionality

The verification tests will start off evaluating behaviour using the culmination merit as the only constraint. This is followed by the addition of other constraints to incrementally build a more complex observational environment. The aim is to always evaluate the expected behaviour against the generated queue. Test results will show the distribution of observations over time. Using the target generator, targets will be constructed in a range of cases set up for easy evaluation.

---

<sup>1</sup><http://rhodesmill.org/pyephem/>

The first test, **case 0**, has three targets that are well separated in HA. This ensures the targets have distinct culmination times. Assuming all observations are of equal importance, the expectation is that scheduling the targets for observation at culmination will result in the targets simply following in sequence. Since the queue is evaluated only on the culmination merit it can be expected that the targets will be scheduled such that the observation time is close to the middle of the target track on sky.

The test targets are defined as follows: the first target rising three hours before sunset, the second target rising 130 minutes before sunset and the third target rising 40 minutes before sunset. Targets culminate at elevations of  $65^\circ$ ,  $70^\circ$  and  $55^\circ$  respectively. The duration of observation for the first and last targets will be an hour, while the second target will only be observed for 30 minutes.

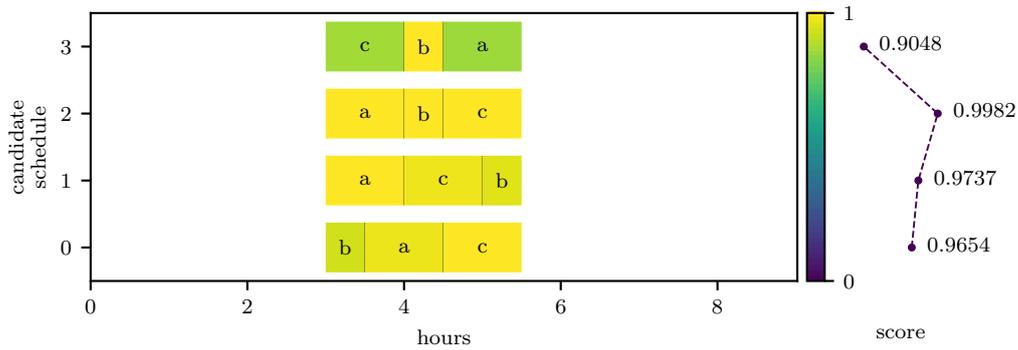
Table 5.1 shows test target coordinates generated using the input time offsets and maximum elevation angles.

**Table 5.1:** Construction of simple sequential targets.

<i>a</i>	( $'10:34:35.03'$ , $'-7:17:34.3'$ ),
<i>b</i>	( $'11:37:52.10'$ , $'-12:17:16.8'$ ),
<i>c</i>	( $'12:29:27.47'$ , $'2:42:40.3'$ )

Using the fake input targets, the greedy algorithm of the scheduler will evaluate a range of observation sequence permutations. Each observation will be allocated an individual score for its position in the queue being evaluated. From this, the queue itself will be assigned a score for the generated observation plan. The queue containing the observation plan with the highest score is then selected.

Visually evaluating the validity of the observation queue selected by the strawman scheduler is easiest if the queue is displayed as a nightly listing of observations, in order, over time. To achieve this style of display, the queue permutations are shown as a bar graph with each observation represented as a block with length equal to the observation duration. The colour of the block indicates the individual observation scores obtained, with the key to the right of the graph. On the far right of the graphic, the respective



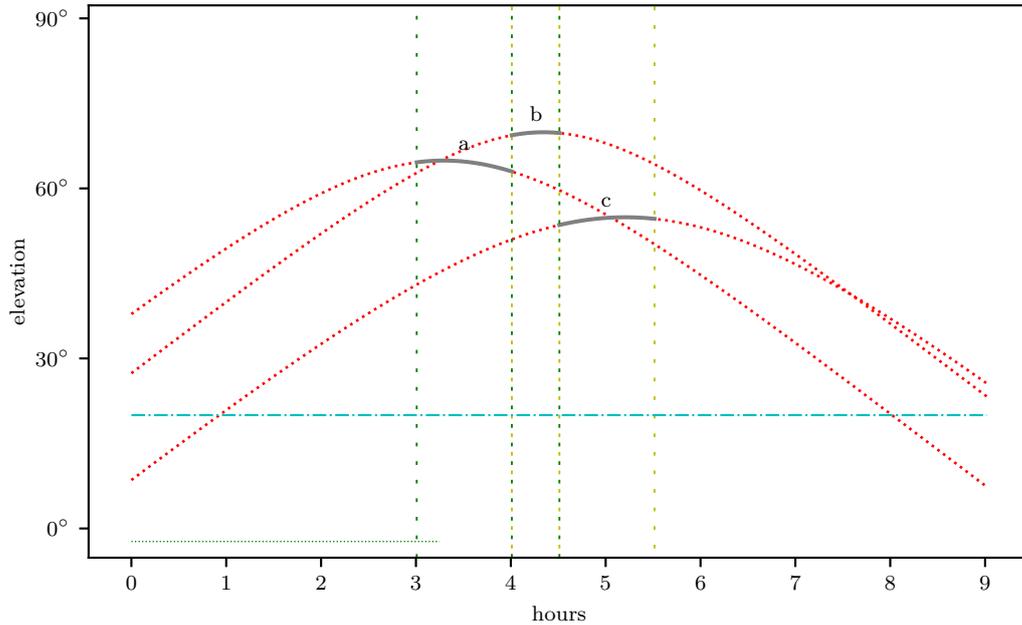
**Figure 5.1:** Evaluating a series of observation queue permutations and selection of the optimal schedule for the sequential culmination observation of the three trial targets listed in Table 5.1.

overall observation scores are graphically depicted—a higher score is shown further to the right. The sequence of observations scheduled in the queue is indicated by labelling the observations *a*, *b* and *c* respectively. Figure 5.1 shows the scheduling results for case 0. Note that the permutations, as shown here, follow in no specific order.

Once the best observation plan has been selected, the queue can be displayed as a function of elevation angle per target over time. Figure 5.2 shows the elevation plot with the sky track of each target as a dotted curve, and the anticipated observation period for each target as a line overlaying the dotted curves. At the bottom of the curve a dot-dash line indicates the horizon-mask, set to an arbitrary constant value of  $20^\circ$  altitude. Below this, there is a shorter dotted line showing the range of start times the `StartTime` algorithm has to its disposal to find the optimal start time for the observation plan.

Candidate schedule two has the highest queue score of the four permutations of queues shown in Figure 5.1 and is selected as the best observation plan to use. The elevation plot in Figure 5.2 shows the target sequence of this queue and it is easily seen that all observation tracks are scheduled at highest elevation for the respective schedule period.

What is interesting is that although the score of the optimal queue, in Figure 5.1, approaches the full score of 1, it is in fact not very close. Expectation would be that, given the simple test setup, the score should be closer to, if not a full score. Additionally,

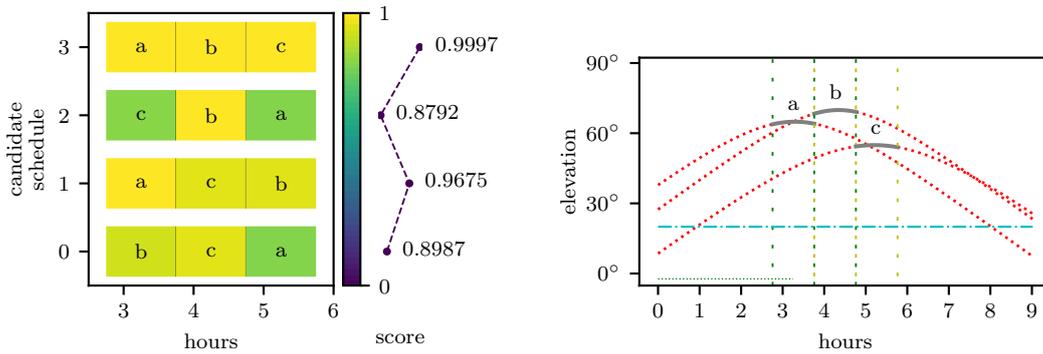


**Figure 5.2:** Generated queue showing sequential targets, scheduled in order, starting from an estimated optimal start time and ensuring all targets are observed close to culmination.

when inspecting Figure 5.2 more closely, it can be seen that the sequence of observations are not scheduled with each observation exactly over culmination.

This highlights a shortcoming in the current implementation of the scheduler. The current implementation of the scheduler is not sophisticated enough to deal with, or to allow, gaps in the scheduling time between target observations. It simply optimises the fill time, or data acquisition factor. This means that observations are scheduled as soon as they can be performed. Changing the scheduling focus to rather maximise scientific output requires adding the ability to allow for non-observation time in order to find the optimal scheduling for the observation. Allowing gaps between scheduled observations is planned as a refinement to be included in a future version of the software. For the purposes of verification, we repeated this test with identical target observation durations, as shown in Figure 5.3. In this case the targets were all observed at culmination.

Even though Figure 5.2 is not the *optimal* outcome, it does clearly show that the obser-



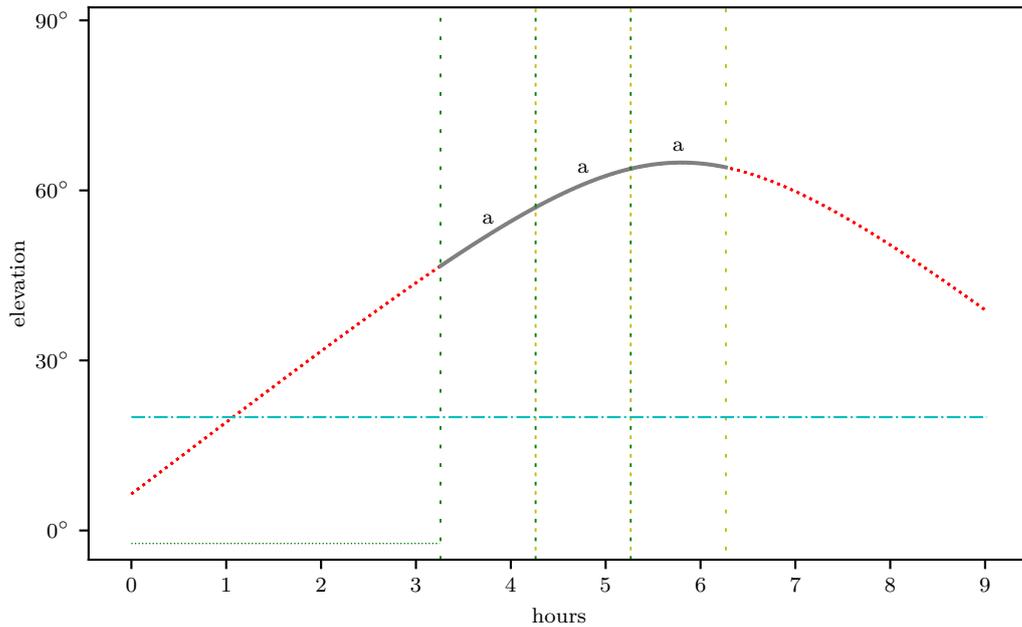
**Figure 5.3:** Updating the duration of observation *b* to better fit the observation period will result in the strawman scheduler providing an improved observation plan. The three test targets are now scheduled to be observed over culmination as expected.

Observations are scheduled over maximum elevation given the restriction of not allowing time gaps between observations. The strawman scheduler still generated a *good* observation queue, given the constraints, showing that the `ForwardPlan` and `LookAhead` algorithms do indeed calculate a valid observation queue. It is however, very sensitive to the time period over which the best start time is selected, and will fail if there are observational time gaps between the targets to be scheduled.

Moving on to create a more complex setup, we include the case where observations compete for oversubscribed HA ranges. This is generally when the TAC priority assignment is expected to influence the observation scores in such a way that the more scientifically interesting targets are favoured.

A very contrived test case is constructed to best illustrate the impact of adding priority to the culmination constraint. Instead of having three distinct targets, as in test case 0, the verification for test case 1 uses a single target, namely ('13:04:57.92', '-7:17:32.5'). The target now has to be scheduled three times, making impact evaluation easier given the expectation that the observations will simply be scheduled in sequence.

Test case 1 sets the priority for all three observations to be equal. When instructing the scheduler to only optimise for culmination and exploiting the fact that the scheduler will fill time starting as soon as possible, the observations are set up to obtain a queue where the last observation is over culmination, as shown in Figure 5.4. This asymmetry

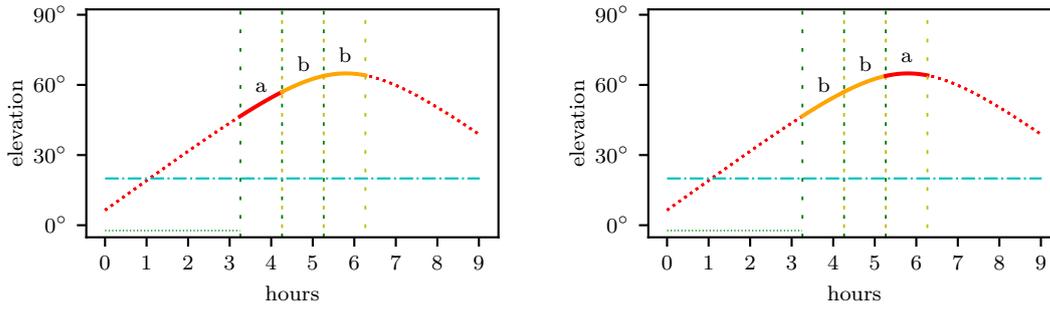


**Figure 5.4:** Scheduling a single target multiple times, with a fixed period in which to evaluate the best start time to get an asymmetric observation queue.

is necessary for visual inspection during the next step, since there is no clear way to distinguish the order in which the repetitions are scheduled.

Test case 2 sets the priority of one the repetitions in test case 1 to be higher than the other two. This has no scientific meaning, but it will demonstrate the impact on the selected schedule, thereby providing a way to evaluate which of the two alternative implementations of the priority merit would be preferable: the priority merit included in the sum evaluation, or as part of the fairness weight.

Figure 5.5 shows the first implementation evaluating the scientific priority simply as part of the merit sum in the graph on the left. The second implementation shows the outcome of the schedule evaluation when the priority is added as part of the fairness weighting shown in the graph on the right. The resulting queue from the first implementation, Figure 5.5 left, is unexpected—the expected result was that both selected observation plans should look like Figure 5.5 right. This result can be explained by looking at the score function definition of Equation 3.1. The merit sum implementation is calculated as an average to prevent artificially favouring observations as more



**Figure 5.5:** The graph on the left shows the priority evaluated as part of the merit sum calculation in the observation score function, while the graph on the right shows the priority as fairness weighting for the three observations. For clarity, the lower priority repetitions are designated as *b*, while the high priority observation with *a*.

merit constraints are added. However, this also has another effect that is inherent to the averaging function, which is to raise values lower than average to be nearer to the average value, but also to lower values higher than the average to be closer to average. When implementing the priority merit as part of the sum, a high priority at an optimal observation position could in fact result in an overall lower observational score, which is undesirable.

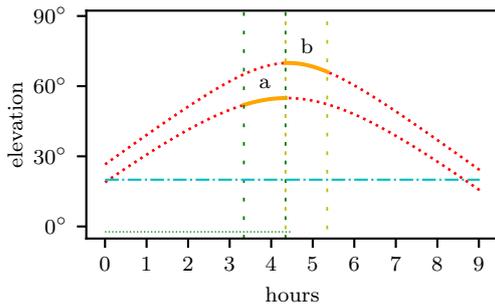
Consider the following simple example: Let the culmination merits for the 3 sequential observations in Figure 5.4 be  $0.1$ ,  $0.5$  and  $0.9$  respectively. Note, these values do not represent the actual values, but serve for illustration. Secondly, let the priority weight assigned to the selected repetition be  $0.7$ . If the priority observation is assigned to the first observation position in the queue, the observation merit sum will update to  $0.4$ , the average between  $0.1$  and  $0.7$ . For the second and third repeats the values will be  $0.6$  and  $0.8$  respectively. The important thing to note is that the observation that would be over culmination and thus having a high score,  $0.9$ , will get a downward correction when the high priority is added to the average,  $0.8$ . Using these observation scores and calculating the per queue score for each of the permutations give the results listed in Table 5.2.

Thus, for the definition of the score evaluation given in Equation 3.1, the more robust implementation of the priority is as part of the fairness weight outside the merit

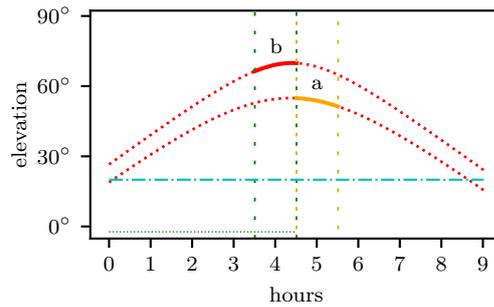
**Table 5.2:** Observation queue evaluation with priority in merit sum calculation.

sequence: a, b, b = $\text{mean}(0.4 + 0.5 + 0.9) = 0.600$
sequence: b, a, b = $\text{mean}(0.1 + 0.6 + 0.9) = 0.533$
sequence: b, b, a = $\text{mean}(0.1 + 0.5 + 0.8) = 0.467$

evaluation.



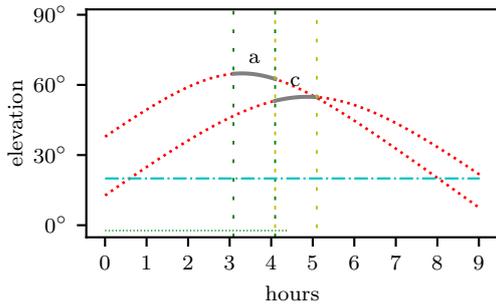
**Figure 5.6:** Different targets of same priority with different culmination elevations,  $b$  higher, but overlapping culmination time. The targets are scheduled in sequence.



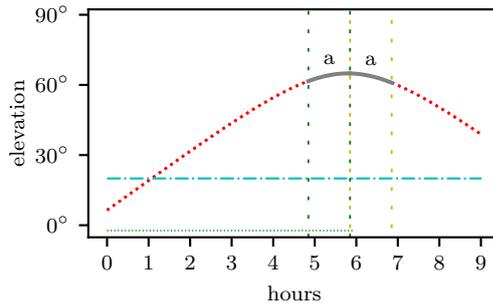
**Figure 5.7:** Setting the priority of target  $b$  higher, causes the scheduler to flip the observation order.

Test case 3 is a slight refinement on the previous test and considers two different targets, but with the same culmination time. If no priorities are assigned the queue shows the targets observed in sequence, Figure 5.6, which is similar to test case 1 and the expected result. While the expectation from test case 2 is that increasing the priority of the target with higher elevation will cause the scheduler to flip the observation order, shown in Figure 5.7.

To simulate how incidental conditions, such as an observation being paused, will effect the schedule evaluation, we update test case 0 such that the observation of the second target is paused and the programme tag for this condition will prevent the observation from being scheduled. We then verify that the first and third target are scheduled just after and just before culmination since gaps are not allowed. This is done both at high elevation and well positioned around culmination, Figure 5.8. For the repeating target setup, simply lower the number of repeats to only twice instead of three times. The scheduler should now position the two observations of the target symmetrically around

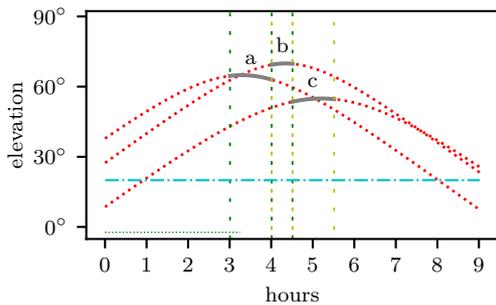


**Figure 5.8:** Regenerating queue of sequential targets with the second target paused and not currently available for observation. The remaining targets are well scheduled around culmination.

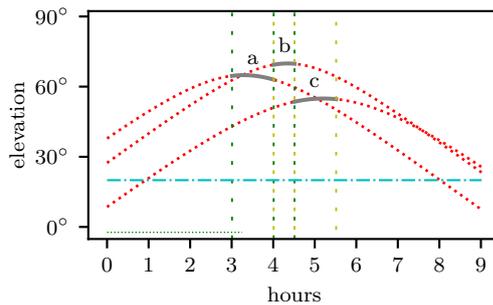


**Figure 5.9:** Changes to observational requirements dropping the number of observations of the same target from three times per night to only twice.

culmination, Figure 5.9. Note that for this case the start time evaluation period was extended to allow an observation schedule symmetrical about the culmination time.



**Figure 5.10:** Scheduling distinct observations optimised using the culmination merit.



**Figure 5.11:** Scheduling distinct observations optimised using the airmass merit.

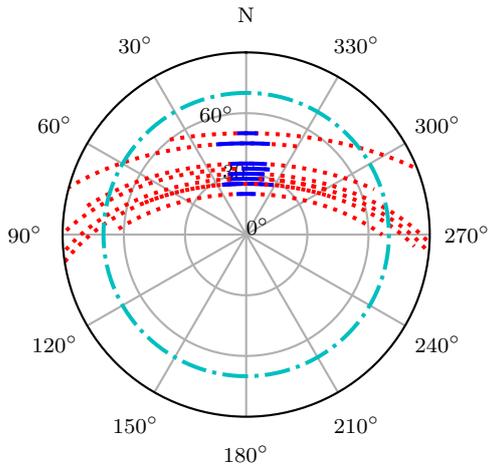
Lastly, the airmass merit should function similar to the culmination merit for the generated test targets since it will drive the observations to have targets with as high an elevation as possible. Figures 5.10 and 5.11 show the elevation graphs scheduling the distinct observations of **case 0** using culmination, Figure 5.10, and airmass, Figure 5.11. These figures compare the result of using the culmination merit compared to the airmass merit. It should be noted that the figures of both cases are identical, as expected.

### 5.3 Observation scheduling

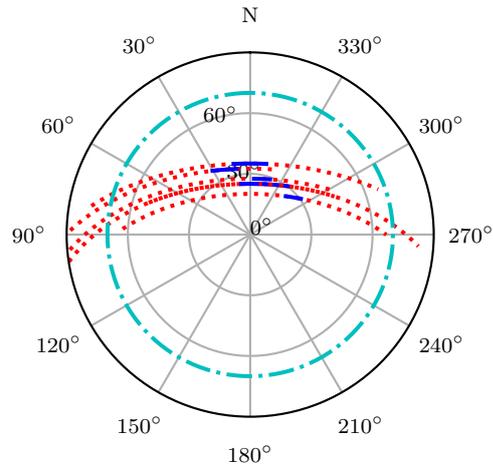
Scheduling optical observations requires the scheduler not only to select relevant observations, but also to optimise open shutter time while filling the night length with viable observations. Since the concept of *night length* is relative to the seasons, the behaviour of the scheduler when building queues for longer (winter), versus shorter (summer) night durations must be validated as well.

To simulate and verify the behaviour of the strawman scheduler when building a queue for a full night of observation, ten targets are constructed to fill a nine-hour time period. For easy visual evaluation the results will again only display the scores when considering culmination versus airmass. The culmination merit will push the observation to be observed over highest elevation, while the airmass merit inherently tries to achieve the same, but by pushing the observations to be observed closer to zenith. The biggest difference will be in the scoring evaluation. While the culmination merit uses a function that has a linear relation between the maximum elevation, with a score of 1 at the transit altitude, to the minimum elevation at the horizon, with a score of 0, the airmass merit follows an inverse cosine function from zenith, with a score of 1, to the horizon value of approximately  $1/40$ . The consequence is that while all targets will achieve a culmination merit of 1, targets with lower elevation culminations will always be assigned a low airmass score. This highlights the need for correct merit selection, but also a need for the scheduler to be robust and not negatively impact the generated queue through introduced biased behaviour. The difference in scheduling is graphically illustrated in Figures 5.12 and 5.13 and discussed below.

The simplest evaluation strategy to select the best queue is to pick the highest average over all per observation scores per queue permutation and to assign that permutation to be the optimal observation plan for the night. When evaluating this best queue selection strategy for culmination, all targets are evaluated over the night and scheduled optimally as shown in Figure 5.14. However, as already highlighted, the airmass evaluation will optimise for zenith angle and thus favour targets that have higher cul-

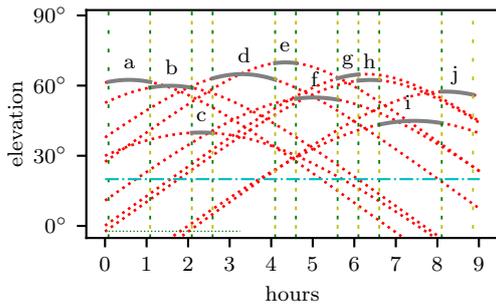


**Figure 5.12:** Polar plot showing the queue of targets all scheduled to be observed over culmination.

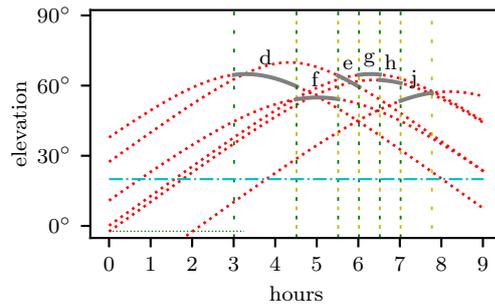


**Figure 5.13:** Polar plot showing the queue of targets all scheduled to be observed over lowest airmass, thus closest to zenith.

mination when evaluated using only airmass. This results in an undesirable queue selection scheduling only the higher culmination targets later during the observation night, shown in Figure 5.15. Again, the high culmination observations at the beginning of the night are lost, due to the current continuous observation time requirements for the scheduler optimisation.



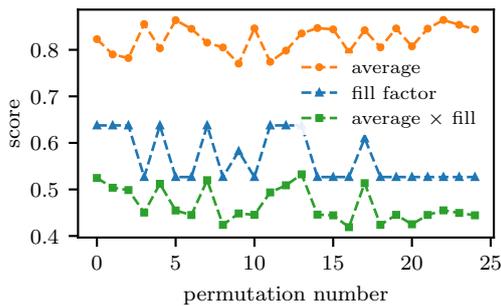
**Figure 5.14:** Observation night scheduling 10 observation over a 9-hour duration, optimised to observe targets at culmination.



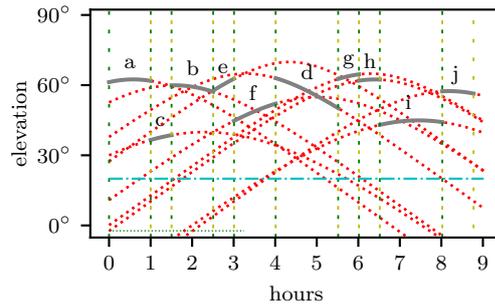
**Figure 5.15:** Observation night scheduling 10 observation over a 9-hour duration, optimised to observe targets at low airmass, using only observation score averaging to select the best queue

Although all targets in Figure 5.15 are scheduled optimally, it is more desirable to obtain perhaps lower elevation observations, with targets throughout the night. Fig-

Figure 5.16 illustrates how the strawman scheduler selection strategy tries to find the highest scoring queue, while maximising the observation time during the night. Here the top, orange graph shows the calculated average over observations per queue permutation evaluated by the scheduler. The centre blue graph shows the corresponding fill factor plotted per permutation. When comparing the average scores per permutation to the fill factor, it becomes clear that the highest average scores coincide with minimal night coverage. Consequently, the strawman scheduler weights the calculated average score with the night fill factor to ensure maximum coverage, even if at a lower calculated absolute queue score, as shown in the bottom green graph. The importance being that the queue scores relative to each other, must be consistent and representative, rather than optimising the absolute score for any single merit. This results in the airmass optimised queue shown in Figure 5.17.



**Figure 5.16:** Queue evaluation options as the average of the observation scores per queue permutation, top orange graph. Queue evaluation by weighting the average, bottom green graph, with a fill factor, blue middle graph, to ensure full night scheduling.

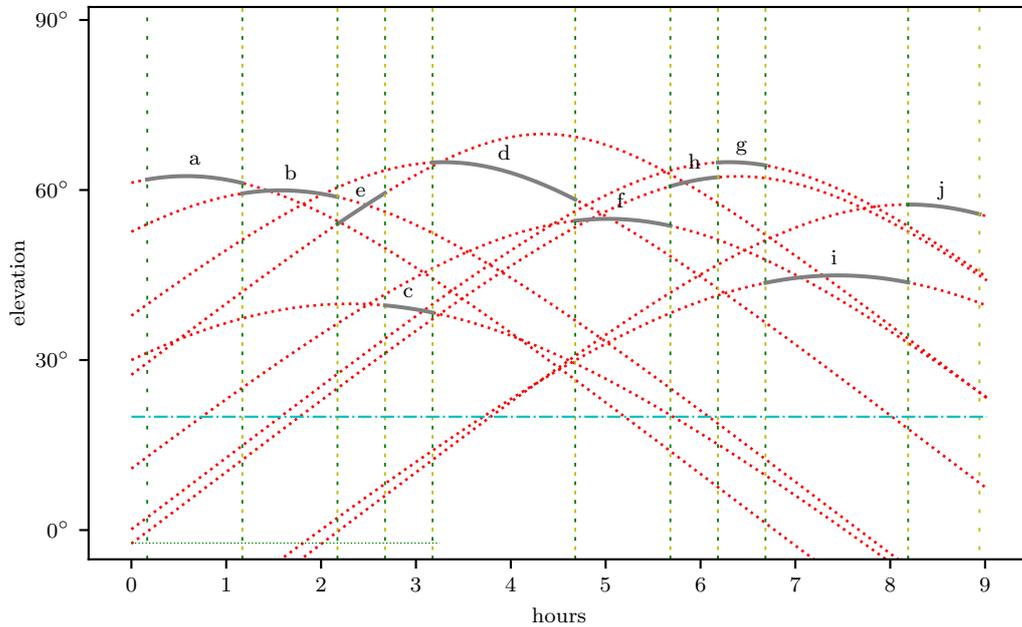


**Figure 5.17:** Observation night scheduling 10 observation over a 9-hour duration, optimised to observe targets at optimised airmass and ensuring maximum night coverage.

Having validated the basic functionality of these merit functions, as well as queue selection evaluations, the last step is to ensure that the scheduler will behave properly over seasonal variations. Both in terms of night length, as well as selected relevant targets to ensure only viable observations are queued.

Utilising the same set of 10 targets, the merit evaluations are selected to be either culmination or airmass on a random basis and evaluated for schedule at start time

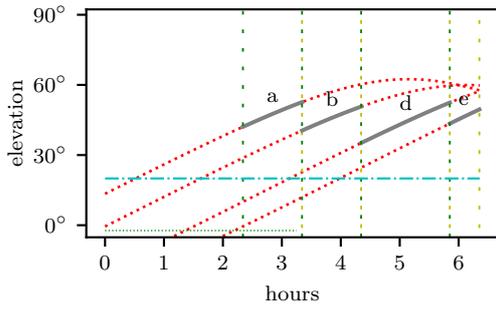
2016/3/16 18:24:46. This results in the observation plan presented in Figure 5.18.



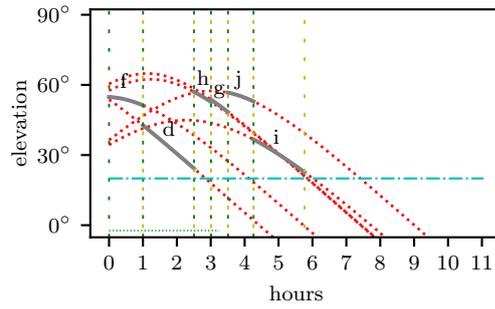
**Figure 5.18:** Nine-hour observation plan for 10 targets with evaluation of either culmination or airmass, randomly selected. The resulting queue fills the observation night and schedules all targets to be observed at high elevation.

To test the queue selection behaviour, the scheduler was set to evaluate queues at summer and winter solstice for the same set of candidate targets. At summer solstice the night starts later and ends earlier, with a night length of only six hours. Also, since the summer solstice is about 3 months before the date used for Figure 5.18, the targets rise later. For the same reason, the targets set earlier in the winter solstice test case. The schedule for the best summer solstice queue is shown in Figure 5.19. At winter solstice, the night starts earlier, and more targets are visible and can be scheduled. With a night length of 11 hours, the scheduler ran out of targets in this test case, Figure 5.20.

It is informative to show the scheduler evaluation for the summer solstice. While five targets were found to be valid for observation over the summer night, only permutations of four targets at a time could be found to be valid over the short night duration, Figure 5.21.

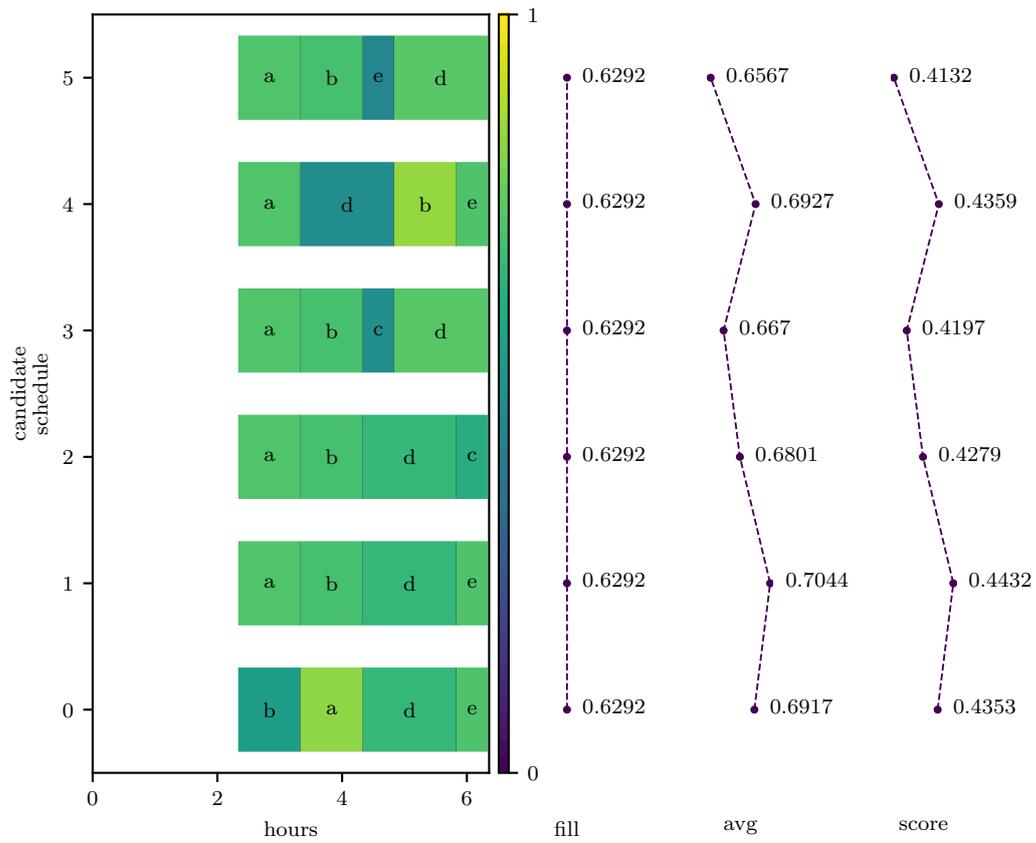


**Figure 5.19:** Observation plan selected for a 10-target list at the summer solstice: 2015/12/20 21:43:42.



**Figure 5.20:** Observation plan selected for a 10-target list at the winter solstice: 2016/6/20 17:06:56.

This chapter highlights how single merit validation tests ensured that the implementation represents the expected behaviour of a human observer successfully. In addition, during the testing phase a number of implementation oversights and oversimplified assumptions has been identified and corrected, resulting in a fairly robust strawman scheduler.



**Figure 5.21:** Short summer nights limit the number of observations that can be scheduled given rise time, per target observation duration, as well as the requirements that all targets be observed at high elevation.

## Chapter 6

# Summary and future work

Planning and scheduling generally refer to off-line processing, while observing requires good solutions with minimal computation time. For each of these stages, we can define the problem input as consisting of the set of observations that have been requested, the constraints peculiar to the instrument/environment, and the optimisation of the objective function (Frank, 2000). This dissertation presents a parametric scheduling strategy to achieve good time distributed observation queues that can be used as an initial scheduler for photometry observations on the ACT telescope at Sutherland.

Maximum science efficiency is achieved by executing the programmes with highest scientific value first, under the required observing conditions. Additionally, maximised scientific use of telescope time is obtained by having appropriate programmes ready for execution under a broad range of observing conditions. The strawman implementation presented in this dissertation exploits one of the easier ways to generate good observation queues by optimising open shutter—on sky—time. This approach simply requires proper time distribution of observations, weighted by scientific priority. In addition, it needs quick and easy evaluation to compensate for queue breakage during observation time, by substituting better suited observations “on the fly”.

Optimisation of choices is essential for astronomical observation scheduling and is achieved by representing constraints as merit functions with a strictness parameter

associated to each. The advantage of using parametric methods is that they are deterministic, which simplifies testing and verification since simple results are expected and results should always be the same for a given observational setup. This helps to build confidence through the construction of schedule outcomes that are predictable given contrived targets specifically generated to allow the input to indicate what should be expected as the generated schedule. In addition, as more constraints are added, they can be evaluated individually as simple single-merit constraints, as well as part of more expanded queues generated, making the implementation modular and easily adaptable.

Having proven the basic functionality of the scheduler, the logical next step will be to verify the queue generation process at a more scientific level. This can be done by comparing computer-generated schedules with human-generated schedules of past observation nights from the ACT. In order to achieve comparable results some future work is required to extend the strawman implementation to allow non-consecutive observation scheduling. By allowing some minimal amount of deadtime between observations, the scheduler will show a preference for selecting higher ranking observations. Thus, by not only focusing on filling time immediately as an optimisation consideration, a science queue generation closer to the natural human evaluation results will be achieved.

Dynamic scheduling can also be introduced by randomly dropping an observation, or moving its location to a fixed observation position. Overflow caused by this simulated meddling requires reworking of the permutations—thereby also influencing the off-line planning—since the induced over-subscription factor has to be absorbed as soon as possible.

# Bibliography

- Bresina, J. et al. (1994). “Automated Management and Scheduling of Remote Automatic Telescopes”. In: *Optical Astronomy from the Earth and Moon*. Ed. by D. M. Pyper and R. J. Angione. Vol. 55. Astronomical Society of the Pacific Conference Series, p. 216.
- Buchner, J. (2011). “Dynamic scheduling and planning parallel observations on large Radio Telescope Arrays with the Square Kilometre Array in mind”. MA thesis. AUT University. URL: <http://hdl.handle.net/10292/3037>.
- Castro-Tirado, A. J. (2010). “Robotic Autonomous Observatories: A Historical Perspective”. In: *Advances in Astronomy 2010*, 570489, pp. 1–8. DOI: 10.1155/2010/570489.
- Colomé, J. et al. (2012). “Research on schedulers for astronomical observatories”. In: *Observatory Operations: Strategies, Processes, and Systems IV*. Vol. 8448. Proc. SPIE, p. 84481L. DOI: 10.1117/12.926899.
- Denny, R. B. (2004). “Dispatch Scheduling of Automated Telescopes”. In: *Symposium on Telescope Science*. Ed. by D. Dais, D. A. Kenyon, and B. D. Warner. Vol. 23. Annual Conference of the Society for Astronomical Sciences, p. 35.
- Frank, J. (2000). “SOFIA’s Choice: Automating the Scheduling of Airborne Observations”. In: *Proceedings of the 2nd NASA Workshop on Planning and Scheduling for Space*.
- Frank, J. and E. Kürklü (2003). “SOFIA’s Choice: Scheduling Observations for an Airborne Observatory”. In: *Proceedings of the Thirteenth International Conference*

- on Automated Planning and Scheduling (ICAPS-03)*. Ed. by E. Giunchiglia, N. Muscettola, and D. S. Nau. AAAI, pp. 226–235.
- Fraser, S. and I. A. Steele (2004). “Robotic telescope scheduling: the Liverpool Telescope experience”. In: *Optimizing Scientific Return for Astronomy through Information Technologies*. Ed. by P. J. Quinn and A. Bridger. Vol. 5493. Proc. SPIE, pp. 331–340. DOI: 10.1117/12.551380.
- Genet, R. M. (2011). “History of Robotic and Remotely Operated Telescopes: The Fairborn Observatory 1979–1989”. In: *Telescopes from Afar*. Ed. by S. Gajadhar et al., p. 1.
- Gómez de Castro, A. I. and J. Yáñez (2003). “Optimization of telescope scheduling. Algorithmic research and scientific policy”. In: *A&A* 403.1, pp. 357–367. DOI: 10.1051/0004-6361:20030319.
- Gorman, T. (2006). *Bad CaRMa - Simple Talk*. URL: <https://www.red-gate.com/simple-talk/opinion/opinion-pieces/bad-carma/> (visited on 2017-07-20).
- Granzer, T. (2004). “What makes an automated telescope robotic?” In: *Astronomische Nachrichten* 325, pp. 513–518. DOI: 10.1002/asna.200410270.
- Johnston, M. D. (1988a). “Automated observation scheduling for the VLT”. In: *ESO Conference on Very Large Telescopes and their Instrumentation*. Ed. by M.-H. Ulrich. Vol. 2, pp. 1273–1282.
- (1988b). “Automated Telescope Scheduling”. In: *Coordination of Observational Projects in Astronomy*. Ed. by C. Jaschek and C. Sterken, p. 219.
- (1989). “Knowledge-Based Telescope Scheduling”. In: *Knowledge Based Systems in Astronomy*. Ed. by A. Heck and F. Murtagh. Vol. 329. Lecture Notes in Physics, Berlin Springer Verlag, p. 33. DOI: 10.1007/3-540-51044-3\_15.
- Johnston, M. D. and H.-M. Adorf (1992). “Scheduling with neural networks—the case of the Hubble Space Telescope”. In: *Computers & Operations Research* 19.3–4, pp. 209–240. ISSN: 0305-0548. DOI: 10.1016/0305-0548(92)90045-7.
- Kubánek, P. (2008). “Genetic algorithm for robotic telescope scheduling”. MA thesis. University of Granada. URL: <http://arxiv.org/abs/1002.0108>.

- Kyte, T. (2009). *This should be fun to watch...* URL: <https://web.archive.org/web/20170330162509/http://tkyte.blogspot.com/2009/01/this-should-be-fun-to-watch.html> (visited on 2017-07-20).
- Littman, M. L. and S. M. Majercik (1997). “Large-scale planning under uncertainty: A survey”. In: *International Workshop on Planning and Scheduling for Space Exploration and Science (IW PSS 1997)*. Oxnard, CA.
- Maartens, D. S., P. Martinez, and R. Van Rooyen (2017). “Automated scheduling for a robotic astronomical telescope”. In: *Proceedings of SAIP2016, the 61st Annual Conference of the South African Institute of Physics*. Ed. by S. Peterson and S. Yacoob, pp. 313–318.
- Martinez, P. et al. (2002). “The automatic photometric telescope at the South African Astronomical Observatory”. In: *Monthly Notes of the Astronomical Society of South Africa* 61, pp. 102–111.
- Mora, M. and M. Solar (2010). “A Survey on the Dynamic Scheduling Problem in Astronomical Observations”. In: *Artificial Intelligence in Theory and Practice III*. Ed. by M. Bramer. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 111–120. DOI: 10.1007/978-3-642-15286-3\_11.
- National Institute of Standards and Technology (2016). *NIST Time Frequently Asked Questions (FAQ)*. URL: <https://www.nist.gov/pml/time-and-frequency-division/nist-time-frequently-asked-questions-faq#cut> (visited on 2017-01-31).
- Pickering, K. A. (2002). “The Southern Limits of the Ancient Star Catalog and the Commentary of Hipparchos”. In: *DIO* 12, pp. 3–27.
- Steele, I. A. and D. Carter (1997). “Control software and scheduling of the Liverpool Robotic Telescope”. In: *Telescope Control Systems II*. Ed. by H. Lewis. Vol. 3112. Proc. SPIE, pp. 222–233. DOI: 10.1117/12.278824.
- Street, R. A. et al. (2018). “General-purpose software for managing astronomical observing programs in the LSST era”. In: *Software and Cyberinfrastructure for Astronomy V*. Vol. 10707. Proc. SPIE, p. 1070711. DOI: 10.1117/12.2312293.

- Swanson, K., J. Bresina, and M. Drummond (1994). “Robust telescope scheduling”. In: *Proceedings of 3rd International Symposium on Artificial Intelligent, Robotics and Automation in Space (i-SAIRAS)*. JPL, Pasadena, CA, pp. 347–350.
- Van Heerden, H. P. (2011). “Robotic control of a photometric telescope”. MA thesis. University of Cape Town. URL: <http://open.uct.ac.za/handle/11427/11399>.
- Verderame, P. M. et al. (2010). “Planning and Scheduling under Uncertainty: A Review Across Multiple Sectors”. In: *Industrial & Engineering Chemistry Research* 49.9, pp. 3993–4017. DOI: 10.1021/ie902009k.
- Wall, M. B. (1996). “A genetic algorithm for resource-constrained scheduling”. PhD thesis. Massachusetts Institute of Technology.