SALT Scheduling Simulations Study with 2014-2 statistics

Paul Kotze, Petri Vaisanen and Astro Ops, Darragh O'Donoghue and Chris Coetzee

May 12, 2015

This document provides the background analysis and motivation for the Recommendations and Requests to the Board presented in the Chapter 2 of the SALT Operations Report by Chris Coetzee et al. This document consists of the following sections, which can be read individually depending on interest. The technical details are described at some length in section 2 and for those with less interest in this, it is possible to skip to section 3 where the results and recommendations are presented.

1. Introduction and motivation
2. Description of the Simulations
3. Summary of the essential results of the Simulations and Recommendations to the Board
4. Statistics of the (real life) 2014-2 Semester

1. Introduction

Since the initiation of full science operations with SALT in 2011, it has not been clear to SALT Operations what performance metric of the telescope should be optimized. Possibilities include (this is probably not an exhaustive list):

- Minimizing telescope idle time
- Maximizing the number of P0-P2 blocks with acceptable science data
- Maximizing the number of all blocks (P0-P3) with acceptable science data
- Maximizing program completion
- Ensuring the observation statistics for each partner follow the partner's shareholding (which we shall refer to as the "equitable time distribution").

Part of the uncertainty on which metric to choose is that reliable statistics on the real performance of the telescope were not available from the start of full science operations in 2011, and have only recently given a clear picture of the performance of the telescope.

As mentioned in the letter written by Mike Shara, Darragh O'Donoghue and Petri Vaisanen, program completion rates have been about 20 per cent; completion of P0-P2 blocks have been about 60 per cent, while the total time used for blocks with acceptable science data as a fraction of available science time is now over 85 per cent.

It is thus opportune for a study to be carried out in which simulations are used to examine what comes out of the telescope in terms of the above performance metrics given different strategies for scheduling observations by the telescope.

The purpose, therefore, of this report is to provide the results of the study to the SALT Board to enable them to declare to SALT Operations how the Board wants the telescope to be operated.

2. SIMULATING OBSERVATIONS WITH SALT

2.1 Source of Input Information

The scheduling simulations make use of actual proposal data submitted for the 2014-2 semester. Around 1200 blocks were submitted in phase-2 adding up to ~900 hours across all priorities. The simulations, coded in Python, also make use of real weather information from the database. It takes around 30 seconds to run one semester making Monte Carlo simulations with hundreds or thousands of runs possible.

2.2 Methodology and Assumptions

The scheduling simulations were designed to closely resemble the current observing strategy used by SALT astronomers. Typically, a nightly queue is built by picking the highest priority target available at the start of an evening, followed by the next highest priority target and continuing in this fashion until the end of the night. This is essentially a “greedy algorithm” which aims to make the locally optimal choice at each stage with the hope of finding a global optimum. Additionally, the SALT astronomers, in practice, aim to minimize the idle time during the night in order to maximize the number of science blocks observed, while also occasionally deciding to wait for a while for a significantly higher priority block to become visible. One problem has been that over previous semesters there never were any clear rules of how to weight the blocks against each other, with the same TAC priority class. In the simulations, if many blocks in a given time window have the same priority or score, the observed block is chosen randomly.

The SALT simulations aim to investigate the current limitations of the observing strategy described above, and to investigate what can be done to improve the high priority science output of SALT. The rules of choosing a block to execute can easily be changed.

The simulations were designed to produce a near-realistic time breakdown for the time spent on science and time lost to weather at the end of a semester. Engineering and technical time, however, are aimed at the specification of about 10% in total, slightly less than the actual for the 2014-2 listed in Section 4.

To realistically simulate the time lost to weather, a sample of more than 500 nights of weather data was assembled. When building a nightly queue of observations a random night, close to the length of the night for which the observation queue is to be build

from the weather sample, is selected. Bad weather is defined as conditions where the dew point delta temperature is below 2.5 (this is the largest single reason for weather downtime) and the relative sky temperature is below 25 degrees, which indicates heavy cloud cover. Telescope closure due to high wind is rare and not included.

Seeing is also included in the simulation. It is not available for each of the nights included in our weather dataset however. Thus, seeing for a given random night's random weather was approximated from the relation between mean wind direction and mean seeing derived from 500 nights of weather data over the past two years. For a given mean wind direction a random selection from the seeing dataset is drawn where the wind direction is within 20 degrees of the mean wind direction of the randomly selected weather data. The mean seeing and standard deviation is returned to the simulation. At each time-step in the scheduler the seeing is drawn from the following: [mean seeing; mean seeing -1; mean seeing + 1; mean seeing + 2; mean seeing + 3].

The scheduling software evaluates the weather conditions at every time-step during the night and decides whether a new target can be inserted into the queue by evaluating the fraction of time that block would spend in unacceptable weather conditions. If this fraction is higher than 0.1 the block is not inserted and the time is counted as time lost to weather.

Technical downtime is modelled as random events that occur during a night and can last between 10-15% of the length of the night (but every night will not have technical downtime, so the full semester figure will be close to specification). If a block can be inserted to the queue, according to the weather conditions, the fraction of time overlapping with a technical problem event is evaluated. If this fraction is above 0, the block time is logged as technical downtime.

Time spent on engineering is also modelled as random events that take place after a block is observed. A random decision is made where there is a 20% chance of doing engineering after a block has been inserted into the queue. The length of time spent on engineering is randomly chosen from ten values that range between 15 and 40 minutes. A maximum of two engineering slots are allocated during a night. This scheme can be thought of as mirror alignments or engineering/commissioning observations, and again the full semester figure will be close to specification.

Another important number tracked by the simulations is the amount of idle time on a given night. This is defined as time without any reason for not observing, but not doing science either. In the simulations this is time waiting for a next available target, or time after the last observation before twilight starts.

2.3 What the Simulations Produce

The simulations produce a number of outputs that are used to evaluate the parameters used during the simulation of a semester. For every night in the simulation, the queue that was built on a given night is stored. The time breakdown in terms of time spent on science, weather downtime, technical downtime, engineering time and idle time is recorded for every night. The science time spent per priority bin on every night is recorded. The time used for proposals in the queue broken up into their individual priorities are recorded.

It is possible to track these quantities recorded on a nightly basis over the course of a semester. Hence, at the end of the semester, as well as at any given point of the

semester, we know how much time was used for any given program, or for any priority bin. In the end, in the particular simulations presented here, we are interested in extracting the following metrics:

- Fraction of P0 time observed
- Fraction of P1 time observed
- Fraction of P2 time observed
- Fraction of P3 time observed
- Fraction of completed programs
- Fraction (and amount) of idle time

"*Completeness*" of programs can be defined in multiple ways. For the purpose of reasonably caveat-free definition to compare with real Semester statistics, we here define it as:

Any program with more than 90% of its allocated time done ("Prog C90")

In the semester statistics Section 4 we also use alternative definitions.

2.4 Time allocation

The SALT TACs allocate time according to four Priorities, P0, P1, P2, P3. They are given time to allocate in the following fashion: 30% for P0-P1 combined, 40% to P2, and 30% for P3. However, the P3 time allocated is then multiplied with a factor of 1.5, i.e. it is oversubscribed. Below, when we refer to P3 completeness, we give both the P3 time observed compared to the time theoretically available for it, and in parenthesis the fraction of completed P3 blocks in the queue (which is a factor of 1.5 lower). Likewise, the total fraction is compared to the available science time and the number in parentheses is the percentage of actual block time completed in the queue (which is a factor of 1.15 lower). The idle time is a fraction of the available science time. In addition to P0-P3 time TACs may allocate an unrestricted amound of P4 (filler) time.

Since the time actually allocated is based on the expected science time available at SALT (semester length minus all downtime) it hence has been implicitly assumed from the beginning of SALT Operations that all P1 and P2 blocks should get observed, in addition to 66% of P3 time.

We measure completion of priorities according to the amount of time assigned to the TACs. It should be noted that if a TAC has not fully assigned their time, or if a PI has not made full use of their assigned time, it will make the completion fractions lower even if all the blocks in the queue were observed.

As a reference (see Section 4) in 2014-2 we obtained the following completion figures for each priority:

P0	75%	
P1	72%	
P2	45%	
P3	60%	(40% of P3 queue)
Total	58%	(50% of total queue)
Prog C90	34%	
P4	56%	(uncharged, fraction of submitted block time)
Idle time	<5%	(see section 4)

2.5 RESULTS FROM DIFFERENT OBSERVING STRATEGIES

2.5.1 Case 1: Minimizing Telescope Idle Time - The Reference Case

We first test whether the simulation comes close to the real 2014-2 semester when programmed to mimic the real life mode of operation. Most often, though not always, the SAs would simply choose the highest priority target at the end of a previous observation. In the this simulation, using real weather and downtime statistics as described in section 2.2, a new block to be observed is chosen randomly from the highest priority ones available at the end of a previous observations. Idle time is employed only if there is nothing available, in which case the next available block is observed.

The amounts of science time (about 550h used), and the various downtimes from this simulation are very close to the real values of 2014-2 given in section 4, though technical and engineering downtime are slightly less than in reality. The key indicators for this reference simulation of the 2014-2 semester are as follows:

P0	69%	
P1	78%	
P2	50%	
P3	61%	(41% of P3 queue)
Total	60%	(52% of total queue)
Prog C90	30%	
P4	67%	
Idle time	11%	(158h)

These numbers, including the Program completion rate, are remarkably close to the actual values given in section 2.4 above. This gives us confidence that the simulation is working as intended. Idle time (as a fraction of the total night time of the semester) is larger than in reality – however, most of it is accumulated during the end of a night when blocks cannot fit in the time left before twilight while in reality SAs are more flexible and often would push an observation into twilight.

Figure 1 shows the cumulative time breakdown vs. time for one of the semester runs. The semester was off to a slow start weather-wise, but the situation improved in the middle of the semester.

The cumulative priority breakdown over the semester is shown in Fig. 2. It shows the effect of the initial bad weather during the semester after which the P1 and P2 time steadily increased till near the end of the semester where the number of P1 blocks started to decrease due to them getting thin in the queue already.

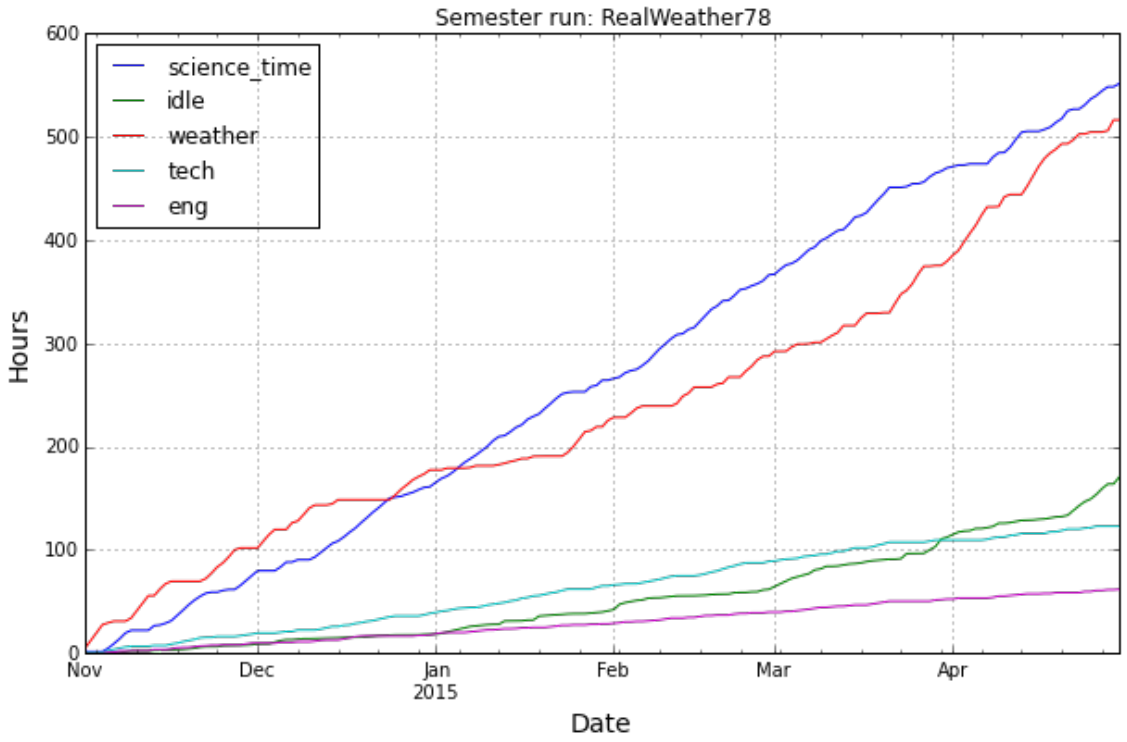


Figure 1: Time breakdown vs. time for a semester run when idle time was minimized and real weather, technical and engineering time was included in the simulation.

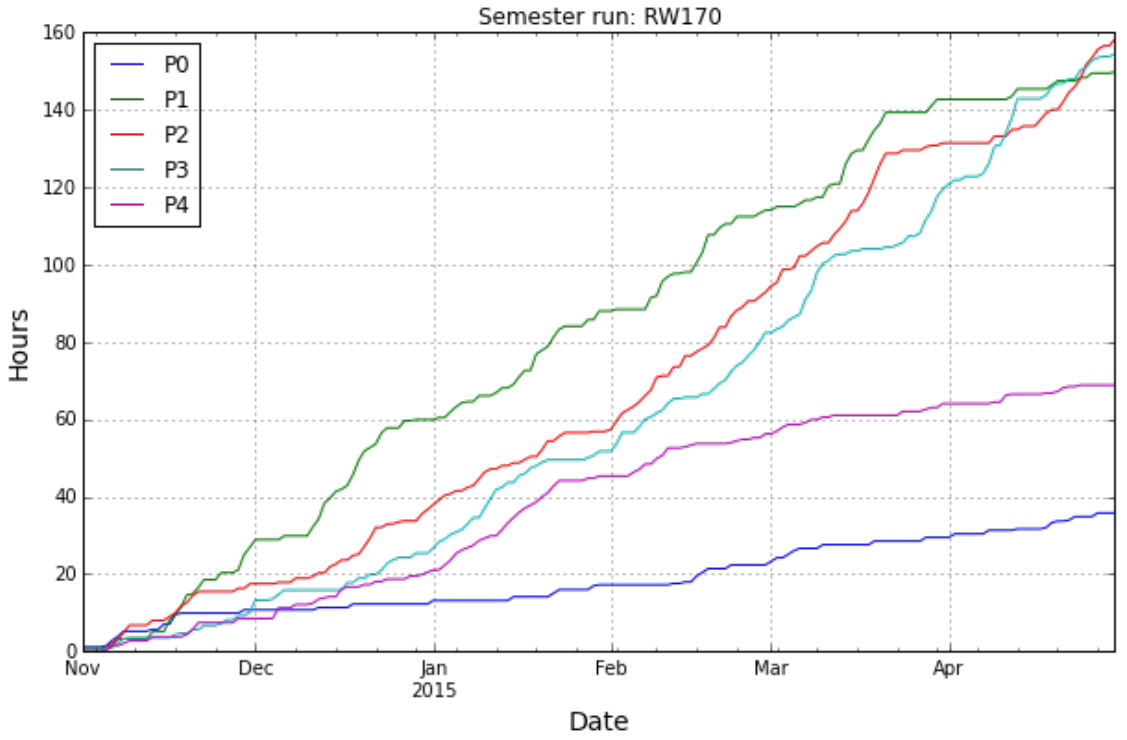


Figure 2: Cumulative priority growth for a semester simulation taking weather, technical downtime and engineering time into account.

2.5.2 Case 2: Impossibility to Complete All Observing Time

Theoretically, we should be able to vastly improve upon the completion numbers above if we have no weather or technical downtime nor engineering time during a semester.

To test this assumption we ran the simulations *without any downtime*. Any departure from 100% time used on science will be unavailability of blocks in the queue to observe. Figure 3 plots the results for used science time forced idle time. The dips in science time at the beginning of each month are Bright time – SALT time was severely undersubscribed for Bright time right from the beginning of the semester[1]. As expected, the number of observable blocks decreases at the end of the semester while the amount of idle time increases. This is not surprising. What is very interesting is to see were all Blocks from the queue completed by the time the queue is essentially empty by April?

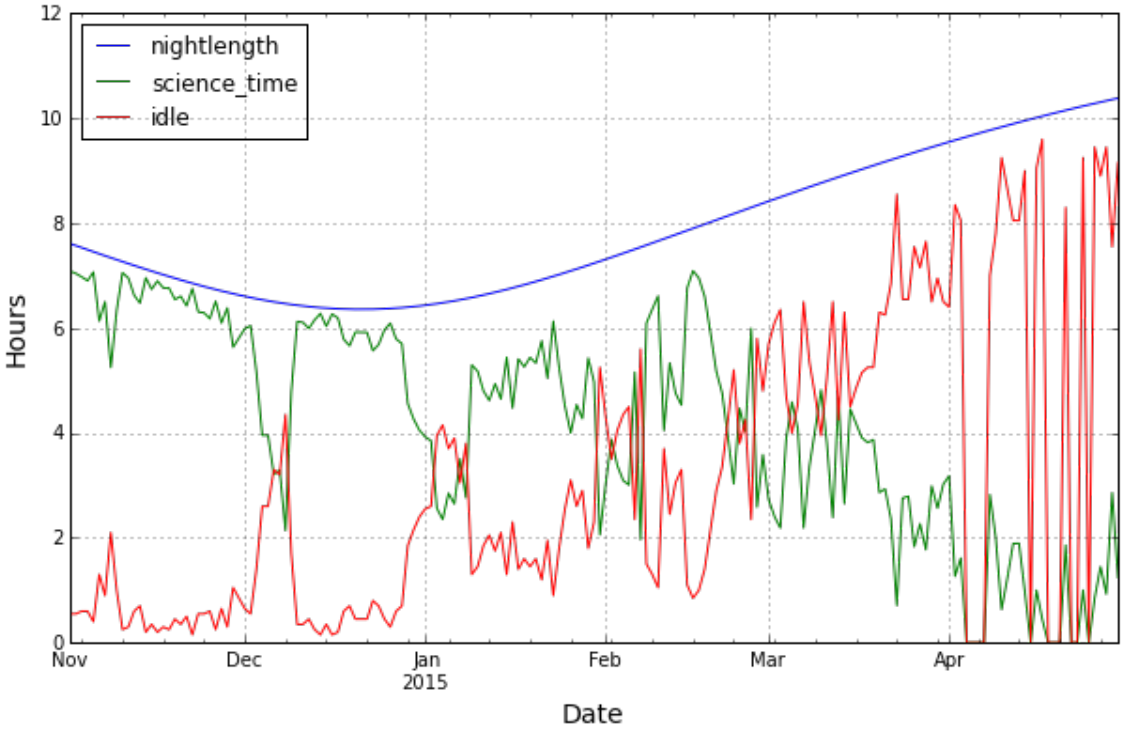


Figure 3: A simulation using the 2014-2 semester proposal information and no weather, technical, or engineering time.

The priority completion breakdown for the simulations, given as an average of multiple runs of the MC semester simulation, is as follows:

P0	82%	
P1	98%	
P2	78%	
P3	90%	(60% of P3 queue)
Total	85%	(74% of total queue)

[1] In the 2014-2 Semester only 3% of time was allocated to Bright time by the TACs. The fractions for Dark, Gray, and Any were 33%, 20%, 44%, respectively. All Any blocks do not cover the Bright Moon phase. Effectively we are allocating time as Dark and Any (see Section 3, point 7).

Prog C90	55%
P4	69%
Idle time	43% (566h)

It is striking that even with perfect conditions we will not be able to finish all the high priority blocks! (The P0 time is not completed because not all ToO programs actually submitted their blocks). One quarter of all the submitted blocks in the queue remain unobserved even if there was no downtime.

The program completeness is especially worrisome, with only 55% of programs getting 90% or more complete.

It is therefore clear that there are other significant limitations to completing programs apart from downtime. Since blocks do not get observed in this case, the reason must be there is too much overlap of the target visibility windows of targets and competition with other blocks. Many DEC ranges are very inflexible with visibility windows and oversubscription of targets there will necessarily mean that only a fraction of them will ever get done.

2.5.3 Case 3: Maximizing the Number of P0-P2 blocks

Next, we do the following: instead of just observing one of the available highest priority blocks at the end of the previous block, or the nearest one, we set a maximum amount of time to look for a higher priority block to become visible. The simulation then waits until the highest priority block within this window. We call this the "idle time mode" where the exact allowable amount can be freely set when building a nightly queue. This method was devised to simulate the case for maximizing the amount of time spent on P0-P2 blocks. P3 gets only done if nothing else is available in the time window. And P4 if not even a P3.

Hundreds of simulations were run using a maximum allowed idle time of 20 and 40 minutes. The resulting average priority completions are:

	20 min	40 min
P0	71%	72%
P1	86%	86%
P2	55%	51%
P3	55% (37%)	50% (33%)
Total	62% (54%)	59% (51%)
Prog C90	28%	26%
P4	66%	63%
Idle time	11% (151h)	11% (165h)

Having an allowed idle time makes an appreciable difference to the overall P1 completion, 86% up from 76% in the reference case of minimizing idle time. This is about 22h more spent on P1 over the semester. That time won effectively comes from the lower priorities, as expected. Surprisingly, however, the overall idle time for these simulations is essentially the same as for the reference case, which attempted to minimize it. We have to look a bit deeper into the simulation to be able to explain that.

One important thing to note is the amount of time, 70h, spent on P4 blocks. The scheduler only “observes” P4 blocks when absolutely nothing else is available. And one must remember that P4 blocks, in away, should not even be there, their time was not allocated at all. The cause is not due to conditions, but just the availability of targets in the queue. Thus, further completion of the higher priorities would be expected if more higher priority blocks were available.

This further shows, as did section 3.3 already for the whole SALT queue, that the on-sky distribution of high priority blocks is not uniform enough to ensure their completion. This can be seen from the growth of the P1 time in Fig. 4 showing the cumulative time spent per priority bin the for the 20 idle time simulation. The staggered increase of the time spent on P1 blocks is an indicator that the P1 block distribution is not uniform as a function of RA during the semester. Moon constraints are the most likely reason for this. An additional reason for the P1 completion not being higher is the fact that not all the P1 time that was assigned for the semester was used, lowering the P1 completion.

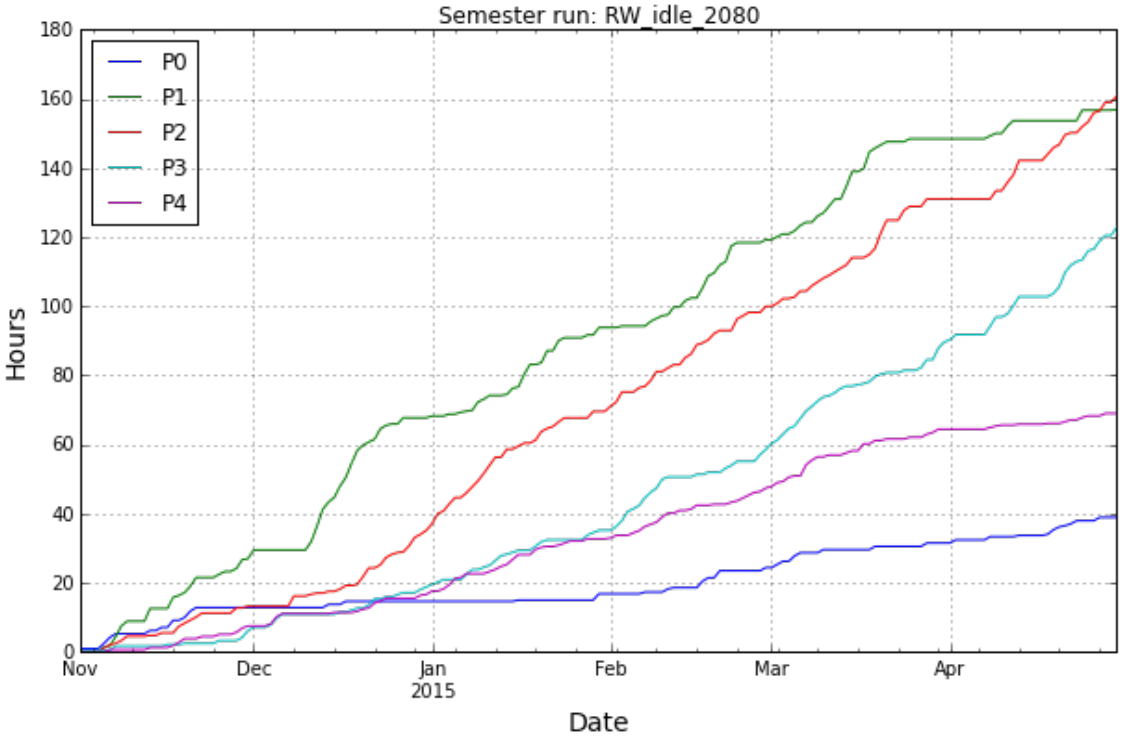


Figure 4: The cumulative time observed per priority over the semester for the 20min idle time simulation.

2.5.4 Case 4: Maximizing program completion

Most proposals consist of multiple priorities. As we have shown so far, it is not possible to obtain completion in all priorities, especially in P2 and P3. It is therefore very hard to fully complete science programs. In an attempt to increase the number of completed proposals a scoring algorithm was implemented. The scoring algorithm contain priority overrides that are meant to aid the completion of programs by boosting the scores of blocks belonging to programs that near their completion, or when the block visibility windows decrease.

The average priority completeness for the scoring simulations which otherwise are the same as the reference case above in section 3.2, and using scoring parameters implemented at SALT in early 2015, are:

P0	69%	
P1	79%	
P2	50%	
P3	61%	(41% of P3 queue)
Total	61%	(53% of total queue)
Prog C90	30%	
P4	68%	
Idle time	12%	(170h)

These numbers are almost exactly the same than the reference case. The scoring algorithm appears to make little difference to the priority completion or the program completeness despite the fact that a higher number of completed programs was expected.

Since the same weather information was used, the same amount of science time was available for the scoring simulations. Figure 5 shows the cumulative time per priority for a semester from the scoring simulations, showing that the reference case in Fig 2 is not identical though the end results almost are. The cumulative P1 time growth is somewhat lower than for the reference simulations. The P2 and P3 growth over the semester are higher, with the time spent on P2 overtaking P1 near the end of the semester to drive more program completion. This could come at the cost of other higher priority blocks, which is the likely reason the overall priority completion numbers are similar to the reference simulation. Because of the continuous competition between blocks the number of proposals completed seems to stay constant regardless of scoring. Again, the fact that proposals are made up of multiple priorities make it difficult to complete programs, even when using a scoring algorithm which theoretically should increase the number of completed programs.

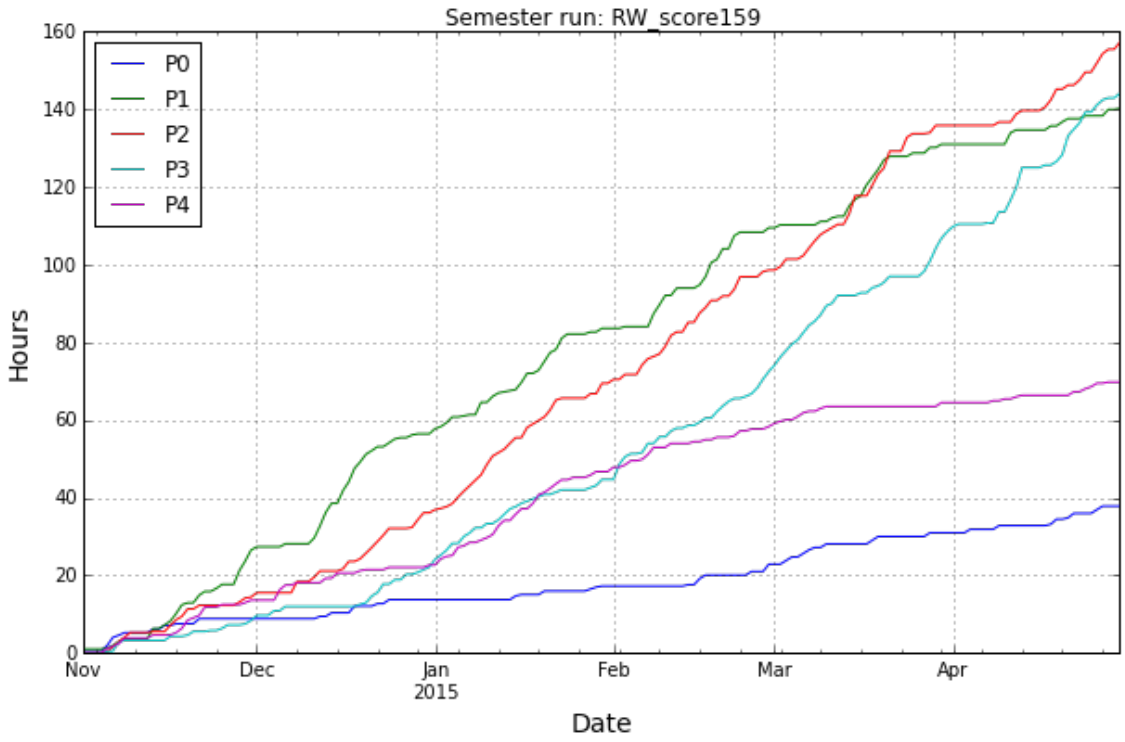


Figure 5: The cumulative time per priority for a semester simulation using scoring. Note the increase in the P3 time over the semester. This is because of the amount of P3 time spread over proposals which needs to be completed to go towards program completion. This does come at the cost of higher priority blocks not being observed.

Importantly, however, the effectiveness of the scoring algorithm is apparent in the number of *single priority* programs completed compared to the reference simulation.

	Reference Sim	Scoring Sim
P1-only programs completed	50%	85%
P2-only programs completed	50%	66%
Overall program completion	30%	30%

More aggressive weighting schemes, in terms of weighting blocks close to completion, were also investigated. Thousands of simulations with randomized weighting were run resulting in program completions ranging between 30% and 40%. Even cases which resulted in ~40% of the number of proposals being completed the P0-P2 block completions fractions were still similar to the reference case. Essentially, the only thing changing is the exact gradient and trajectory of the cumulative Priority graphs over time. Whatever one does, it seems, the program and block competition, mainly based on their visibilities, drives the outcome to similar values.

Inherently, there are certain limitations to successfully observing blocks, and therefore, completing programs:

- Competition between blocks of the same priority
- Restricted visibility windows
- Moon restrictions
- How closely a block fits into its visibility
- And, last but not least, the fact that programs are split into different priorities.

2.5.5 Case 5: What if … ??

In order to get an idea of the effect the visibility windows of targets have on the efficiency of SALT, a number of simulations were run where the visibility window start and end times for blocks were set to the end of evening twilight and the start of morning twilight respectively. We therefore assumed that SALT is a conventional telescope. Block competition will still play a role, but the limitation of having to point to a target at a particular time during the night was been removed.

The priority completion were, for otherwise the reference and scoring simulation:

	Reference	Scoring
P0	80%	80%
P1	91%	94%
P2	67%	69%
P3	65% (43%)	67% (45%)
Total	72% (63%)	74% (64%)
Prog C90	41%	45%
P4	67%	66%
Idle time	5% (64h)	3% (45h)

These represent ~20% increases in the P0, P1 and P2 completions compared to the reference simulation and what was recorded in the past semester. There was a significant decrease in idle time as would be expected, but the program completion still remained around 40-45%. While the observing mode selected for these simulations might not optimize correctly a case of a fully steerable telescope (there is much more flexibility in choosing blocks), the fact that we still did not manage to get even close to overall program completion strongly suggests that the effect of clustered RA ranges and programs consisting of multiple priorities are severely affecting this metric.

We will investigate this in more detail in the future.

2.5.6 Other cases for the future

The cases above present the work done thus far. A few other ideas to test in the future are listed below.

- Effect of ensuring Partner equitable time distribution. We could be expected this to further limit overall program completeness.
- Effect of using large pools of optional targets inside proposals with wide RA ranges. We expect this to be a powerful way to get program completions up while simultaneously minimizing idle time.
- Testing effects of various over-subscription factors. We would expect this to decrease idle time and increase efficiency, but it might not increase program completion, at least by itself.
- Perform true “optimized scheduling” simulations, similar to what currently can be done for a single night using Brent Miszalski’s code, and compare these to the simpler greedy algorithms using scoring scenarios.

- Quantifying which kind of programs get done the best. Some essential tips were already shared in the letter to the TACs by Mike Shara, Darragh O'Donoghue and Petri Vaisanen, in February 2015.

3. MAIN CONCLUSIONS FROM SIMULATING OBSERVATIONS WITH SALT

1. Target visibility constraints are the biggest driver of what can and cannot be done with SALT time currently. Programs cannot be finished even if 100% of a Semester was available, not even all the highest priority Blocks. This is mainly because of conflicts between targets at the same position in the sky, time-restricted targets competing for the same time slots etc.

2. There are not enough targets in the queue. There will be significant idle time of the telescope regardless of which observing strategy is employed with the current queues.

3. SALT observing has been running reasonably close to the theoretical maximum over the past semesters if the purpose is to maximize the number of blocks observed. The completeness percentages appear low; they have improved significantly recently since the attention of Astronomy Ops turned to completing proposals, but they cannot be significantly higher with the current strategic decisions concerning the kind of proposals in the queue.

4. Completion of Blocks of Priority P0-P2, or Partner equitable time distribution, can be driven somewhat higher by employing various Block Scoring algorithms. However, the overall Program completeness rates will likely not improve materially without some changes in top-level strategy addressing issues 1 and 2.

TOP LEVEL STRATEGIC CHOICES

Does the SALT Board want to maximize the number of papers (in peer-reviewed journals) coming out the telescope? Is its highest priority to ensure every partner gets their fair share of observing time? Does it wish to see the highest quality science get preferential priority? Does the partnership wish to guide the allocation of time by their respective TACs? If so, is there a consensus on what guidance to the TACs should be given?

RECOMMENDATIONS

1. It is strongly recommended that, when allocating telescope time, TACs consider visibility distributions, repeat visits to a given target over a short time scale, restrictions on seeing and weather and so on. This will win back significant science time on-sky. If a decision is made, Astro Ops can investigate methods how to do this in practice, including possibilities of obtaining more information about targets and tracks in phase-1 already.

2. It is strongly recommended that TACs allow only a single Priority per Program if program completeness is a goal. This is the only way to increase program completeness significantly.

3. Increase over-subscription of Blocks in queue. The best way would be to heavily use optional targets and pools within proposals and TACs should encourage this. Another option is to increase the time given to allocate per semester – this would help on-sky efficiency, but would hurt overall program completion.

   A best scenario is still under investigation, but something like 40% P1, 40% P2, 60% P3 of available science time should work better than the current scheme, *if* the goal is set to maximize P1 and P2 program completion together with points 1 and 2 above. Larger P1 and P2 share would aid fitting whole program there, while the total oversubscription would increase from the current factor of 1.15 to 1.4. In this case P3s would essentially be used as fillers, but 1/3 of those blocks would still be expected to get complete. Schemes can be utilized to attempt to maximize a certain fraction of P3 programs getting a complete dataset, rather than all of them some data.

4. It is recommend the Board consider whether carry-over of some small carefully determined number of unfinished P1 blocks should be allowed in the interest of striving for highest priority program completion. If the Board thinks this is useful, they should commission a study from Astronomy Operations of the pros and cons of the effect on the queue, what work would be needed, and to come up with any practical ways of doing it, by the next Board meeting.

5. Statistics show that the winter semester has consistently around 45% weather loss, whereas the summer semester is closer to the 35% specification. We recommend changing the time allocation assumption for the winter semester to 45% weather loss.

6. We request clear requirements and goals on what SALT observations actually aim at.

    - Decide what the final goal to optimize is. Programs? P0-P2 blocks? Something else? While recognizing that P3 blocks will not get a high fraction of completeness, aim at completing those P3 programs that do get started?
    - Based on that decision requirements can be set up. If, for example, our recommendations above would be followed including queue shaping with realistic target visibilities, we could suggest the following metrics for science time performance.
    - Meets expectations requirement for a Semester:
        - 90% of P1 blocks completed.
        - 75% of P2 blocks completed.
        - 55% of all programs completed.
    - Longer term goals to aid in further optimization plans:
        - 100% of P1 and P2 blocks completed.
        - 35% of P3 blocks competed.
        - 80% of all programs completed.

7. It is recommended that time allocations move away from the traditional strict Dark, Gray, Bright bins. Experience has shown that a simple Dark and Any split would be more flexible at all levels, in constructing phase-2 blocks the PIs can specify their maximum Moon phase levels. Effectively, SALT time is already allocated this way with 77% of the 2014-2 allocations in these two bins. However, heavily biased distributions in Moon ranges should somehow be smoothed before establishing the final queue (point 1 above).

4. SEMESTER 2014-2 STATISTICS WITH BRIEF COMMENTS

Semester statistics are available from Steve Crawford's pages, including partner and program statistics. The monthly breakdown is repeated below with a slightly different format and with definitions.

https://www.salt.ac.za/statistics/2014-2_stats.html

Observing Statistics for SALT Semester 2014-2							
Month	**Night Length (hr)**	**% Weather downtime**	**% Technical downtime**	**% Engineering Time**	**% Science Time**	**% Obs efficiency**	**% Shutter Open Time ****
Nov 2014	7.07	49%	9%	13%	29%	74%	7%
Dec 2014	6.40	39%	8%	19%	34%	76%	18%
Jan 2015	6.76	28%	12%	11%	49%	92%	27%
Feb 2015	7.84	28%	11%	18%	42%	90%	20%
Mar 2015*	8.69	36%	6%	11%	48%	90%	23%
Apr 2015	10.00	51%	3%	7%	39%	90%	19%
Mean	7.84	35%	10%	14%	41%	84%	19%
Spec		*35%*	*3%*	*7%*	*55%*		

** Shutter open time does NOT INCLUDE HRS yet!

Definitions:

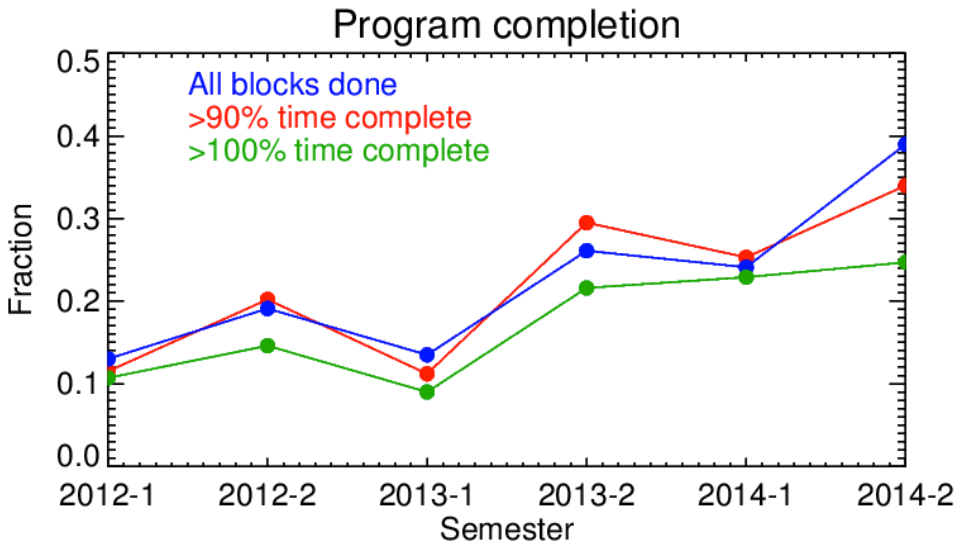
- *Night length* – time from twilight to twilight
- *Science time* – Night length *minus* the sum of the 3 categories below

- *Weather downtime* – any time not on sky due to conditions begin out of spec
- *Technical downtime* – any time lost to problems when weather would allow obs
- *Engineering time* – pre-defined and agree-upon intervention to on-sky time

Note the order of downtime: weather trumps others types. Similarly, otherwise useful engineering time will not be logged under engineering, if it's done only because of a technical problem or weather.

- *Observational efficiency* – Charged time divided by the science time.
- *Open shutter time* – Time spent in science integration (NB: right now the database cannot deal with HRS exposures, so this is not very useful yet).

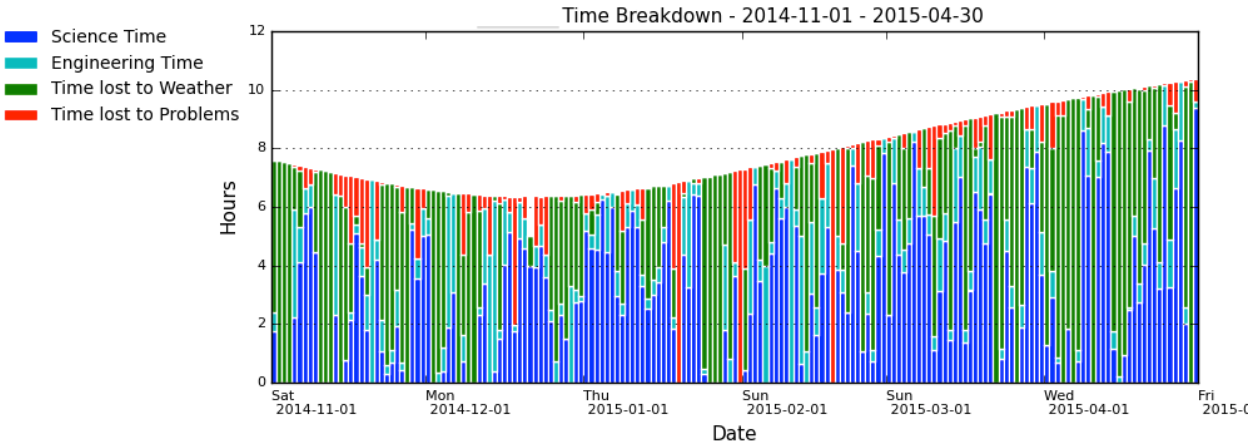
Observational efficiency might need a bit of elaboration: Since charged time is the sum of all the accepted Blocks as defined by the PIs in phase-2 and Science time is the night length minus all downtime, a 100% score would mean all available time was used exactly as predicted for blocks. Less than 100% may mean that acquisitions take longer than anticipated (600s is expected) or there is idle time in between blocks (e.g. while we wait for a higher priority blocks to come visible instead of doing a lower priority one immediately). This statistic is used to track down inefficiencies of the observing as well as check whether time charging is statistically correct.

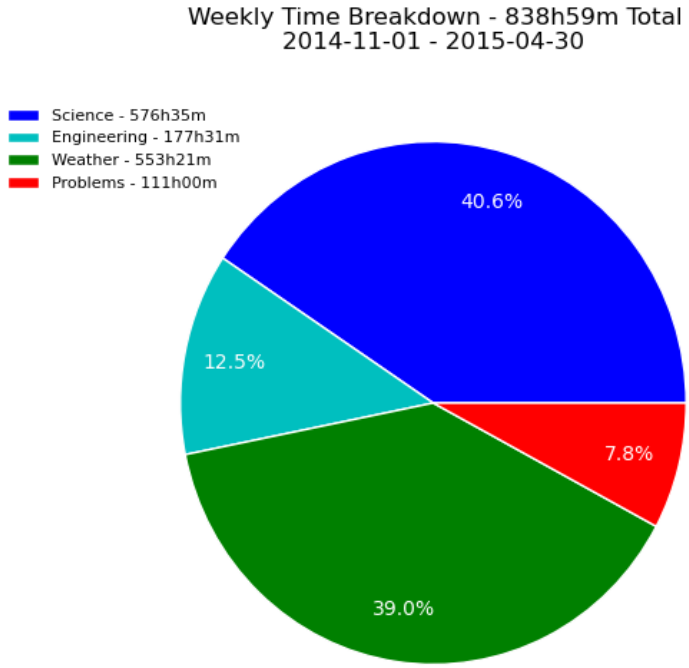


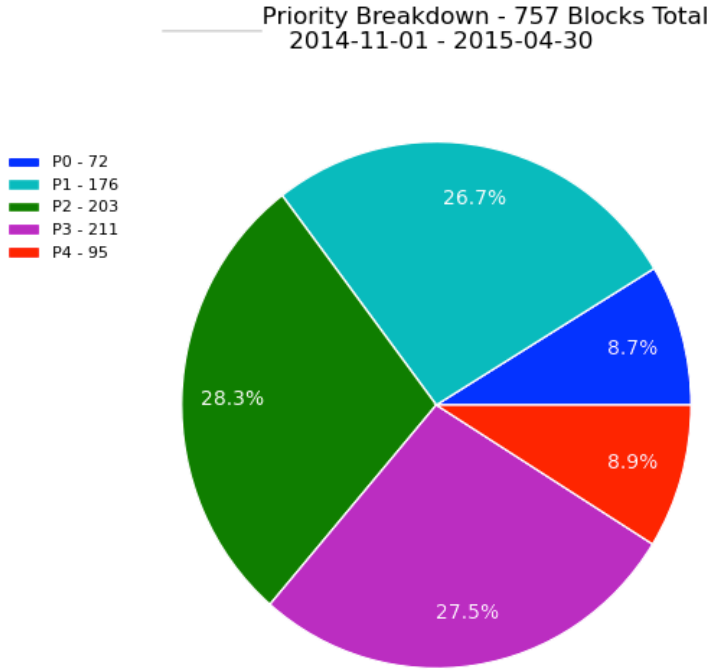
Program completion using various defintions. The up and down trend is due to the summer/winter difference evident from the table above.

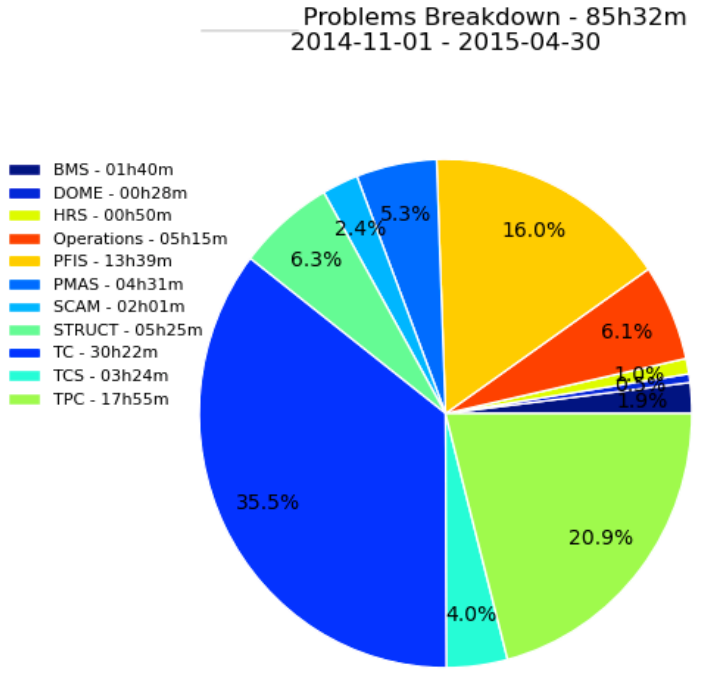
- **Level of 30-40% is at the theoretical maximum reached in the simulations of Section 2**
- **In 2014-2, 63% of P1-only programs were complete, and 83% of P2-only programs. These fraction are similar to those found from Simulations for the Scoring case, the same BIcok Scoring which was implemented during the semester.**

Semester 2014-2 in plots and numbers






Priority Breakdown - 757 Blocks Total
2014-11-01 - 2015-04-30
P0 - 72
P1 - 176
P2 - 203
P3 - 211
P4 - 95
26.7%
8.7%
28.3%
8.9%
27.5%


Problems Breakdown - 85h32m
2014-11-01 - 2015-04-30
BMS - 01h40m
DOME - 00h28m
HRS - 00h50m
Operations - 05h15m
PFIS - 13h39m
PMAS - 04h31m
SCAM - 02h01m
STRUCT - 05h25m
TC - 30h22m
TCS - 03h24m
TPC - 17h55m
5.3%
2.4%
16.0%
6.3%
6.1%
1.0%
1.9%
35.5%
20.9%
4.0%

Some historical stats

Semester	Fraction of Science time	Fraction lost to Weather	Fraction lost to problems	Fraction of engineering time
2011-3	0.41	0.35	0.12	0.12
2012-1	0.38	0.48	0.07	0.07
2012-2	0.47	0.35	0.10	0.07
2013-1	0.36	0.46	0.09	0.08
2013-2	0.45	0.35	0.07	0.11
2014-1	0.32	0.43	0.10	0.14
2014-2	**0.41**	**0.39**	**0.08**	**0.12**
Specification	0.55	0.35	0.03	0.07

Semester	P0 time completeness	P1 time completeness	P2 time completeness	P3 time completeness	Total Time completeness
2011-3		0.41	0.58	0.58	0.53
2012-1	0.49	0.48	0.40	0.28	0.37
2012-2	0.64	0.65	0.56	0.35	0.53
2013-1	0.71	0.74	0.62	0.34	0.51
2013-2	0.77	0.70	0.50	0.32	0.60
2014-1	0.56	0.55	0.36	0.24	0.42
2014-2 (*)	0.75	0.72	0.45	0.40	0.58
Specification	NA	NA	NA	NA	NA

(*) Note that this semester did not have an extra 5% commissioning time taken away from the allocations as did the previous ones, making it that much more difficult to achieve a same percantage as before.

How does SALT time get used? Case of 2014-2.

1420h night time available.
782.7h of time was allocated for the assumed 55% of available on-sky time.
452.9h of this was observed

That is 58% efficiency. Why was 100% not reached?
Taking into account the 39% vs 35% spec weather, 7.8% vs. 3% technical time spec, and 12.5% vs. 7% spec, we still should have observed for 656h (85% efficiency).

So where did that remaining 127h hours go??

- P4 time total: 45.5h.
- That leaves 82h or about 10% of the time that is “un-accounted” for.
- This is a combination of IDLE time (waiting for targets to become visible) or taking more time to execute blocks than charged for.
- Analysis of block lengths from point to next point have consistently given values of 1.05 to 1.10 “undercharging” (however, that is for RSS only, HRS blocks have not been analysed yet). It does seem that most of the “un-accounted” time is in the 5-10% more time spent on blocks than charged for. Possibly some part (<40h) is in IDLE time.

- **This means that our current observational method is decidedly geared towards minimizing idle time.**