



# WFIRST Scheduling Studies: Year One Status Report

---

Mark Giuliano<sup>1</sup>, Reiko Rager<sup>1</sup>, George Becker<sup>2</sup>, John Debes<sup>1</sup>, Anton Koekemoer<sup>1</sup>, Christopher Hirata<sup>3</sup>, Jeffrey Kruk<sup>4</sup> and Roeland van der Marel<sup>1</sup>

## **Abstract**

Studies showing that the proposed WFIRST observing programs are feasible with respect to scheduling are a key component of STScI's effort to support the WFIRST mission. This report presents the first year's system development accomplishments and scheduling study results. Observation specifications were created that implement the core WFIRST science goals as well as representative General Observer (GO) programs. Constraint calculation routines were created leveraging existing prototype tools that create scheduling constraints for the specified observing programs. A software system that is capable of planning and scheduling these observation programs was created leveraging existing STScI software infrastructure. After describing the observing programs and software infrastructure, initial scheduling results are given showing that the system can efficiently schedule WFIRST observations, and can be tailored to meet the specific needs of the WFIRST mission. While the observing program and software systems represent most core mission features, each is an initial version and is missing components. As such, the report ends with an outline of work required to make a complete prototype scheduling system.

## **1 Introduction**

A series of scheduling studies were performed in fiscal 2015 as part of STScI's effort to explore facets of the WFIRST mission. These efforts are built on top of a similar effort done by the WFIRST Science Definition Team (SDT) as described in their 2015 Report (Spergel et al., 2015).

The long term goals of scheduling studies are to show that the WFIRST program goals are feasible with respect to scheduling:

- Establish that the core science objectives for the mission are mutually compatible
- Quantify the expected scheduling efficiency

---

<sup>1</sup> Space Telescope Science Institute, Baltimore, MD

<sup>2</sup> University of California, Riverside, CA

<sup>3</sup> Ohio State University, Columbus, OH

<sup>4</sup> NASA Goddard Space Flight Center, Greenbelt, MD

- Inform engineering specifications and trades:
  - The impact of detector quality on scheduling feasibility
  - The impact of a geosynchronous orbit versus an L2 orbit
- To inform the planning and scheduling architecture to be used in an operational WFIRST planning system.

The studies performed in Fiscal 2015 worked toward the long term goals. A prototype planning and scheduling system was defined and exercised using a geosynchronous orbit. At the time the studies were performed, a geosynchronous orbit was the baseline for the mission. However, at the time of writing the results, the baseline orbit changed to L2. It was decided to publish what was found so far and to perform future L2 based scheduling studies in subsequent years. Also note that while the studies are designed to inform an operational planning and scheduling system, the architecture described below should not be construed as a proposed planning and scheduling architecture for the WFIRST ground system.

The studies were informed by a multi-institute, multi-disciplinary team whose roles are described below:

- Scheduling constraints and Algorithms:
  - WFIRST domain experts
    - Jeff Kruk (GSFC)
    - Chris Hirata (Ohio State)
  - Scheduling Software development
    - Mark Giuliano (STScI)
      - Software architecture, scheduling algorithms, performing scheduling studies
    - Reiko Rager (STScI)
      - Scheduling constraints, observing program specification, and visualization tools
- Strawperson observing program creation
  - Anton Koekemoer (STScI) - HLS imaging & spectroscopy, SN survey, Overall interface between science (and SDT) and scheduling study development team
  - George Becker (STScI) – GO programs – Has now left STScI
  - John Debes (STScI) – Coronagraph observations
- STScI WFIRST Mission Office and Liaison to GSFC WFIRST study office
  - Roeland van der Marel (STScI)

The team has regular twice a month teleconferences to discuss issues and coordinate efforts. These scheduling studies build on and extend a prototype WFIRST scheduling system built by Chris Hirata. Hirata's code was used to help create the observing programs and to calculate WFIRST observing constraints.

The remainder of the report covers the following topics:

- The strawperson set of observing programs created for the scheduling study
- The software systems developed for scheduling WFIRST observations
- The results of the scheduling study
- Future studies and scheduling enhancements

## 2 Strawperson Observing Program

The main science programs for WFIRST are:

- High Latitude Survey (HLS)
- Supernova search (SN)
- Microlensing (ML)
- Coronagraphy (CG)
- General Observer (GO)

Detailed observation specifications were created for all of the programs except for the microlensing program. The microlensing program is handled by blocking out 6 ~72 day intervals for observing the single field. Details on the 6 observing epochs are given below.

The scheduling system developed for WFIRST plans *visits* or groups of exposures that are designed to be scheduled in a single block of time. The groups of exposures may iterate through a set of filters and/or perform a mosaic over a small pointing range. A visit has a duration which includes both the time on target for each exposure as well as the overhead time for intra-visit slews and filter moves. Schedule efficiency measures reported below are for the single duration and do not attempt to quantify intra-visit overheads.

### High Latitude Survey

The High Latitude Survey (HLS) is a contiguous region covering 2227 deg<sup>2</sup> consisting of imaging in four near-IR filters (Y, J, H, and F184), as well as grism observations covering the wavelength range 1.35 – 1.89  $\mu$ m. The observations are arranged in a series of contiguous “fields”, each corresponding to the full coverage area of the WFIRST imaging camera. Imaging observations for each field are obtained in two passes, with 3 to 4 exposures per pass, dithered by  $\frac{1}{4}$  SCA offsets to cover the chip gap. The second pass, at a different roll angle, completes the depth and provides additional sub-pixel sampling as well as observations of sources at different locations on the focal plane to help with calibration. The fields are also covered with grism observations, obtaining these in 4 passes at different roll angles, with 2 exposures per pass. For this scheduling study, all the exposures were grouped into “visits”, where a given visit generally consists of all the dithered exposures necessary to observe one field, in all filters, during a single pass. This structure enables efficient scheduling of exposures that are very close together on the sky, as the  $\frac{1}{4}$  SCA offsets are most effectively handled internally to each visit. Adjacent fields, and also second-pass observations, are then handled as separate visits.

### Supernova search

The supernova search consists of two target locations near the ecliptic pole, which are monitored using the Wide Field Imager for a total of 6 months across a 2-year period, with visits repeated every 5 days. The observations for this scheduling study were grouped into a total of 288 visits (144 visits for each of the two pointing locations),

distributed throughout the 2-year observing season. During each 5-day window, there is some flexibility in how the visits can be scheduled, with the only constraint being that each group of visits be completed during each 5-day window. As a result, the visits are generally schedulable throughout the 2-year season, with minimal impact from constraints from avoidance angles due to their location near the ecliptic pole.

## **Microlensing**

The microlensing target area is observable twice a year, each approximately 72 days long, providing 12 seasons in the 6 year planned mission. Six of these seasons, the first 3 and the last 3, are used for microlensing observations. Since they provide concrete scheduling blocks on the mission timeline, we have decided against modeling and scheduling individual observations. Rather, any time the microlensing target is observable in the six chosen seasons, no other observation is scheduled. When the moon avoidance constraint makes the microlensing target un-observable, the scheduler is allowed to schedule observations from other programs.

## **Coronagraphy**

A specific design reference mission for the Coronagraphic Instrument (CGI) has currently not been defined, but will primarily be used to observe nearby stars to directly image the reflected light from exoplanets. The primary mission consists of the detection and characterization of nearby exoplanets detected through radial velocity surveys, a blind search for new planets, and the characterization of debris disks, all of which were included in our strawperson programs and based as closely as possible on details covered in the 2015 SDT report. The total mission duration for the CGI is roughly one year and will likely cover a range of targets scattered throughout the sky. Long exposures (up to 10 days), especially for spectroscopic observations, are expected.

## **General Observer Program**

It is impossible to predict what targets and observing setups GO programs will request by mission start. Therefore, the primary goal of the GO programs is to provide a mix of observations that cover a variety of instrument modes, coordinates, and observing strategies in a manner timely enough to be useful for the scheduling study, rather than direct relation to the actual science goals a rigorous GO science program might take. Care should be taken when using these mock GO programs for other purposes. The GO programs were primarily designed from a combination of the JWST NIRCAM SODRM (for the WFI), the WFIRST SDT one-pagers (Appendix D in the 2015 SDT Report), and a mix of likely targets for coronagraphic GO programs including directly imaged planetary systems, detailed follow-up of newly imaged planets, quasar hosts, and known resolved circumstellar disks.

More details on each strawperson program are provided in Appendix A, including the plots of the target locations.

Table 1 lists the total visit duration and the number of visits in strawperson programs, as well as what are specified in the SDT 2015 report. Since the total GO programs do not use as much time as specified in SDT report, we have doubled the number of non-coronagraphy GO programs by cloning each observation with the target coordinate flipped. With the addition, we have the total of 11384 visits taking 5.77 years, not including inter-visit slews, for the planned 6 year mission timeline.

<b>Program</b>	<b>SDT 2015 – Section 3.10 (The numbers include all overheads)</b>	<b>Strawperson Programs (The numbers include intra-visit overheads but do not include slews between visits)</b>	<b># visits</b>
HLS	2.01 years	588.98 days	4771
SN	0.63 years	192.56 days	1440
ML	72 days x 6	362.7 days (Excludes the un-usable time due to the moon near the line of sight)	N/A
CG	1 year	375.27 days	332
GO	≥ 1.25 years	314.15 days (113.22 days based on JWST SODRM + 113.59 days based on SDT appx. D + 85.90 days GO coronagraphy)	2466
Total	6.07 years	5.08 years	9009

**Table 1. Total duration of observations from each science program.**

Each science program poses different scheduling challenges:

- HLS restricts scheduling to a fixed position angle restriction for each observation in order make the mosaic tiles line up;
- Supernova observations are repeated every 5 days preventing long observation from being scheduled in between;
- Microlensing observations reserve large blocks of time in the first and last one-and-half years of the 6 year mission timeline;
- Coronagraphy has many very long observations of more than 10 days;
- The GO strawperson programs contain a mixture of different programs; long observations, short observations, observations that need to be repeated at varying position angles, observations that need to be repeated after certain amounts of time.

Our aim is to see if we can create a schedule that accomplishes the main scientific objectives and also to see if we should make recommendation as to what kinds of GO programs should not be accepted.

There are several differences between the work presented here and that done by Hirata and presented in the 2015 SDT report. Hirata adopted a given HLS and SNS implementation, and showed that it can be scheduled with enough time left to do GO and CG science. Here instead we actually develop strawperson GO and CG science programs, and we actually schedule them jointly with the same HLS and SNS implementation adopted by Hirata. Moreover, Hirata's schedule served primarily as an existence proof. By contrast, our software approach allows us to flexibly explore different scheduling approaches for the same observations, so as to optimize merit functions that can be specified to the software.

### 3 Scheduling Architecture

Figure 1 shows the high level WFIRST scheduling system architecture developed for WFIRST planning and scheduling. The architecture was designed to leverage existing software infrastructure including:

- WFIRST astronomical constraint calculations coded in Chris Hirata's system
- Planning, scheduling, and visualization available in STScI's SPIKE planning and scheduling tool kit.

SPIKE is a planning and scheduling tool-kit that was developed at STScI for planning HST observations and is under development for JWST (Johnston and Miller, 1990; Giuliano and Johnston, 2010).

#### 3.1 WFIRST Astronomical Constraint Calculations

Chris Hirata's prototype tools were modified to allow for the calculation and disk storage of WFIRST astronomical constraints including sun, moon, earth constraints, and spacecraft roll restriction constraints. Hirata's program is documented in (Hirata, 2015).

Our modifications to the software included:

- Encapsulating the calculation of the WFIRST position vector so it can be calculated once and used to calculate constraints for multiple visits
- Wrapping the code so it can be used as a shell script

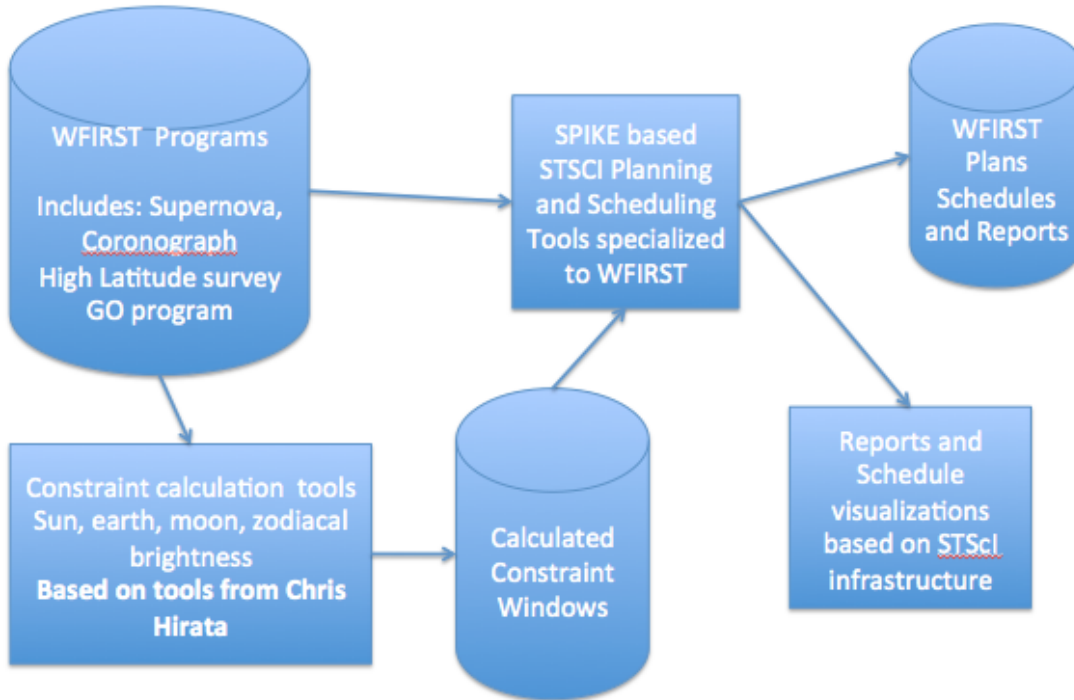
At the time of this report writing, the constraint calculation has the following simplifications:

- All observations have the constraints calculated as WFI observations (i.e. target placed on the center of the WFI apertures), even when calculating constraints for coronagraphy observations which are centered on a different location on the focal plane.
- Constraints are calculated for every 1/10 day (= 2.4 hour) and every whole position angle degree (0, 1, .. 359). The *schedulability* windows are created by combining the time and position angle sample results in a conservative manner.

The schedulability windows are thus up to 1/5 day shorter than what should have been. For visits that did not have schedulable windows because the target is affected by the Earth occultation constraint and the visit's duration makes the visit not fit within any intervals between Earth occultations, we have calculated schedulability windows using a smaller sampling interval (1/100 of a day). If a visit still did not have schedulability windows, it was broken up into multiple visits.

- Zodiacal light restriction constraint is not currently used.

## Scheduling Architecture



**Figure 1: WFIRST scheduling architecture.**

Visualization tools were created to help analyze the scheduling of individual observations as well as understanding WFIRST scheduling constraints. Figure 2 shows scheduling constraints for a coronagraphy observation over a two year time span. The plot shows when the observation is schedulable and the particular constraints that combine to make the observation unschedulable over other time periods. Figure 3 shows the legal spacecraft roll for a target over time. These tools provide support for the creation of science programs that achieve science goals while being schedulable. Likewise these tools can help analyze unschedulable visits. In addition, we have calculated the effect of the sun, moon and earth constraints for a 5x5 degree grid of targets over a year time span. Plots from the calculated data are in Appendix B. This visualization provides coarse-grained input to proposal creation. For example, the plot implies that a program using a target at the ecliptic may not be able to use more than ~94 days in any one year. 5x5 degree target plots were also calculated showing the target visibility for each day over a

years time span. These individual plots can then be integrated into a single movie showing how target visibility moves over the course of a year.

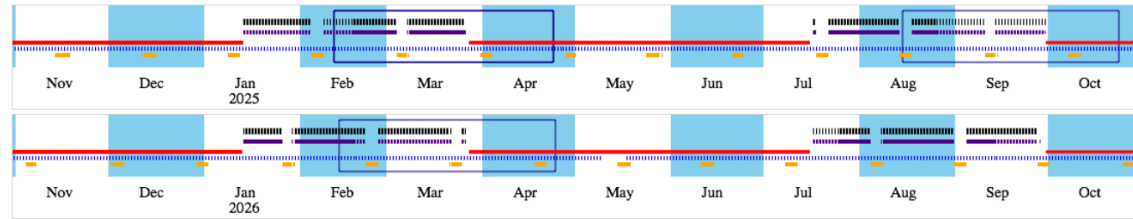
### Program 70000 - Coronagraphy (half of 225 obs) GEO (6/6)



\* Gaps shorter than 8 hours may not be reflected in the plot.

V70000\_0194

(Ra, Dec) = ( 56.583, 26.216), duration: 21600 seconds = 6.00000 hours



**Figure 2: Scheduling constraints for a visit over a two year window**

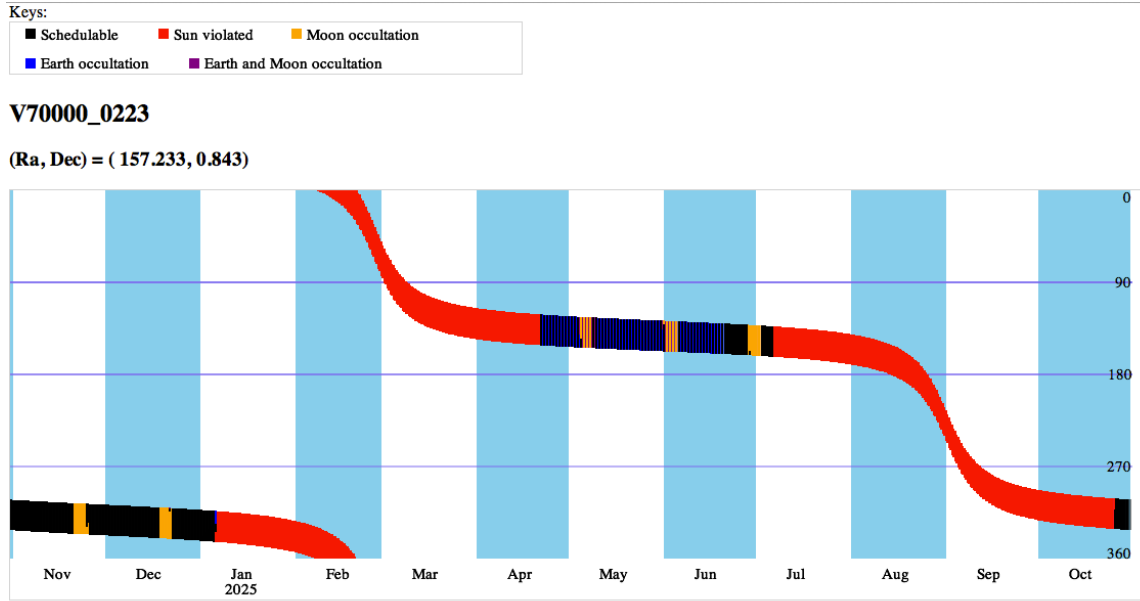
The constraint calculation tools support a geosynchronous orbit as well as a L2 orbit. Current studies have only exercised the geosynchronous orbit model.

The goal for the first year of the study was to capture the key scheduling constraints for the mission in order to determine mission feasibility. Several constraints were not implemented due to the lack of time or performance concerns. Table 2 summarizes what was implemented and what was not implemented. It is important to note that the SPIKE planning and scheduling system provides infrastructure for all of the unimplemented constraints.

Scheduling Constraints Implemented	Constraints not Implemented
<ul style="list-style-type: none"> <li>- Geosynchronous orbit Sun, Earth, Moon avoidance</li> <li>- Roll availability – Is there a roll which can be held the duration of the visit?</li> <li>- Roll restrictions – User specified roll restrictions</li> <li>- User time restrictions</li> <li>- Slews between observations</li> </ul>	<ul style="list-style-type: none"> <li>- Timing links between observations</li> <li>- Roll links between observations (e.g. orient OB1 from 10-20 degrees from OB2)</li> <li>- Guide stars</li> <li>- Zodiacal light</li> <li>- Assigning a roll for all observations</li> </ul>

**Table 2: The scheduling constraints that are currently implemented and the constraints not implemented**





**Figure 3: Constrained spacecraft roll for a visit over a year**

### 3.2 Inter-visit slew and settle times

Hirata’s prototype code provides routines to compute slew and settle times given a pair of right ascension, declination and position angles. Our first plan was to call Hirata’s C code from SPIKE which is written in LISP. We have decided against that idea because calling an outside function frequently during scheduling will significantly slow down the scheduling process. It also forces us to run SPIKE only on Linux machines, where Hirata’s code is installed, whereas SPIKE can be run on other platforms otherwise. Instead of calling Hirata’s code or converting the entire routine from C to LISP, we have decided to use approximate slew times based on slew angles. The slew angle from a pointing to another pointing is the angle of rotation around the rotation axis. Figure 4 shows the relationship between slew angles and computed slew times on randomly generated data. The estimated slew duration to use in SPIKE is based on median values on every 0.5 degree slew angle range.

Slew angle (in degrees)	$0 \leq x < 0.06245$	$0.06245 \leq x < 1.249$	$1.249 \leq x < 5.534$	$5.534 \leq x \leq 180.0$
Settle time (in seconds)	6.5	8.5	13.25	25.25

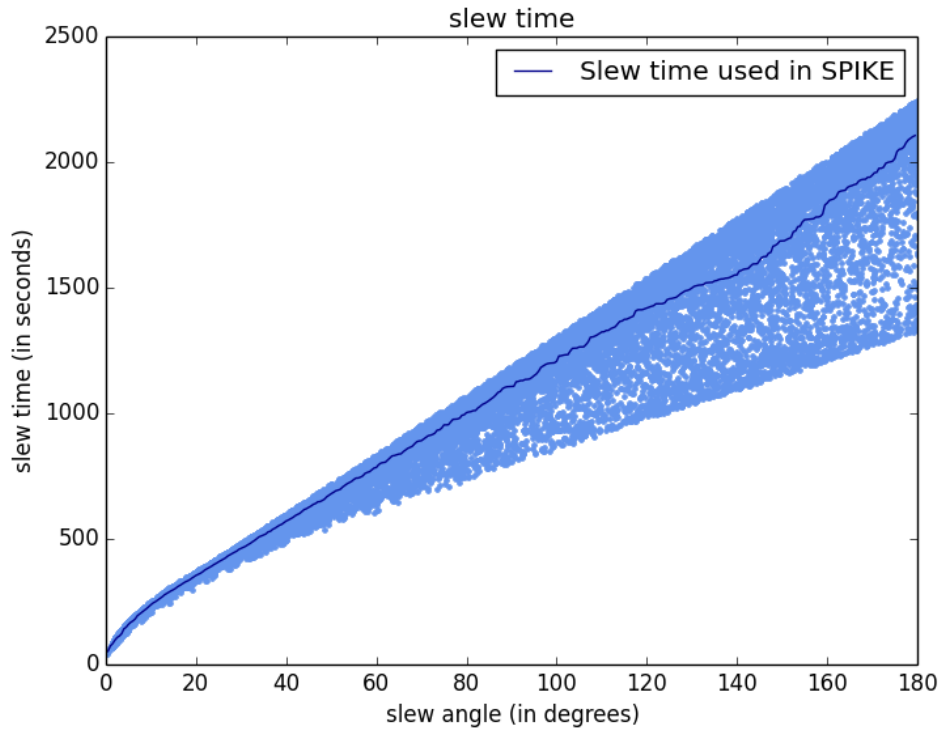
**Table 3: Settle times in seconds for given slew angle**

Table 3 provides the settle times used in SPIKE, which include the 0.25 seconds of guide star acquisition time. The values are straight from Hirata’s code. According to his note (Hirata, 2015), settle times are based on the simulations by E. Stoneking.

The estimated slew time based on median values seem to work adequately well. Using 5 different schedules created by SPIKE, we have compared the slew-settle times used there against the slew-settle time as calculated using Hirata's code. The results as shown in Table 4 shows the differences between the total slew duration, summed over the entire schedule, is only about  $\sim 1\%$ , with SPIKE using slightly higher values.

Schedule	A	B	C	D	E
Total slew-settle time over entire schedule from Hirata's code	5322951	4737824	5457402	5303239	4795703
Total slew-settle time over entire schedule used in SPIKE	5389620	4787364	5511420	5362199	4852150
Difference of total	1.25%	1.05%	0.99%	1.11%	1.18%
Mean of the differences in individual slews	6.162	4.559	4.859	5.313	5.188
Standard deviation of differences in individual slews	68.645	781.08	833.246	822.245	748.09

**Table 4: Comparison of slew-settle times between Hirata's code and SPIKE**  
(duration is given in seconds)

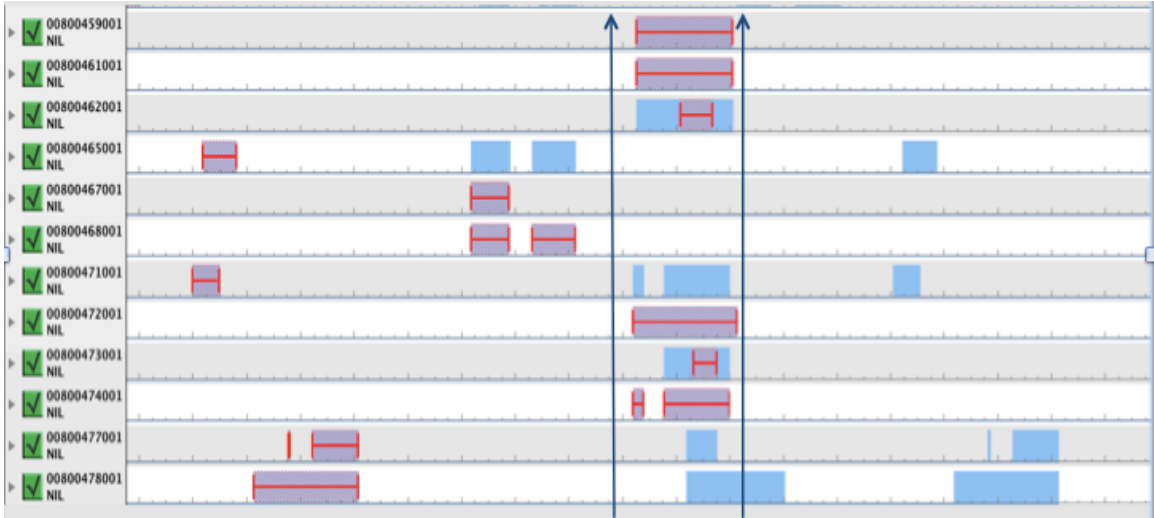


**Figure 4: Slew time of randomly generated pairs of pointings. The line indicates the median values that are used as the estimated slew duration in SPIKE**

### 3.3 WFIRST Planning and Scheduling Engine

#### 3.3.1 High Level Architecture – Long Range Planning and Short Term Scheduling

The SPIKE system provides a tool kit consisting of multiple scheduling engines supported by a rich model of astronomical constraints. The system supports planning and scheduling at different time granularities. For both HST and JWST, planning and scheduling is a two-phase process. In the first phase, long range planning creates coarse-grained least commitment plan windows that are nominally 56 days long. Short term scheduling takes plan windows as input and schedules observations to precise times. Plan windows are a subset of an observation’s constraint windows and represent a best effort commitment by STScI operations staff to schedule within the window. Figure 5 gives an example of a long range plan. Each row represents a visit in the plan. The light blue windows give the constraint windows for the observation while the red bars show a selected plan window for each visit. The short term scheduler will then create a detailed schedule where all visits are scheduled to precise times within their plan window. For example when scheduling the interval between the two horizontal arrows in Figure 5 the short term scheduler will only consider those visits with overlapping planning windows.



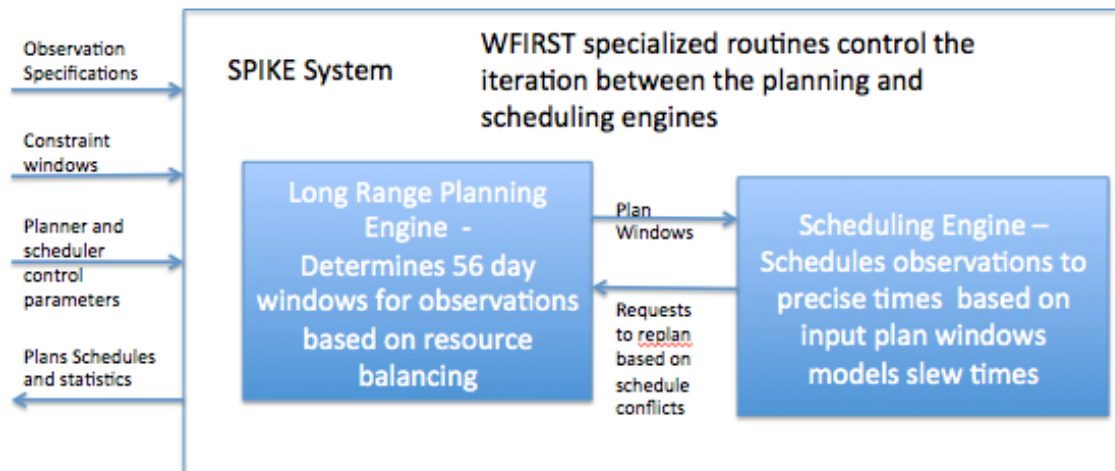
**Figure 5: Visits with constraint windows and a long range plan**

This two phase process allows a separation of concerns in the scheduling process: Plan windows globally balance resources, are stable with respect to schedule changes, and provide observers with a time window so they can plan their data reduction activities; Short-term schedules provide efficient fine grained schedules to the telescope, and handle slews and other activities between visits.

For WFIRST the main motivation for the two-phased approach is runtime efficiency. In principle one could just run the short term scheduler without first creating a long range plan. To test the feasibility of this approach, a short term scheduling engine was created for the observation set without considering restrictions from a long range plan. After

creating the short term scheduling engine, a simple search algorithm was started to test the efficiency of the system. Measuring the rate at which observations were scheduled, it would take about a week of CPU time to complete one scheduling run on a server class machine. In contrast, running a long range plan then a series of short term schedules takes on the order of 5 hours on a relatively underpowered laptop.

The SPIKE scheduling architecture for WFIRST is given in Figure 6. The architecture tightly integrates long range planning with short term scheduling in a single running image. This allows iteration between components. Both the long range planner and the short term scheduler have pre-existing scheduling algorithms that were specialized and/or scripted for WFIRST. In the remainder of this section descriptions are given of long range planning procedures, short term scheduling, engine procedures, and procedures iterating in between the long range and short term planning engines.



**Figure 6: SPIKE integrated planning and scheduling architecture**

### 3.3.2 Long Range Planning Engine

The SPIKE long range planning system assigns visits to 56 day long plan windows with the goal of balancing resources and optimizing planning criteria. The planning engine uses a two-step guess and repair strategy. The planner first assigns all visits to plan windows attempting to optimize the resource usage for each visit. After assigning all visits a repair step attempts to move visits to undersubscribed times. The high-level procedure can be put in pseudo code as:

```

Procedure long-range-plan(visits, criteria)
    Initial-guess(visits, guess-candidate-criteria(criteria), guess-window-criteria(criteria))
    repair(visits, repair-candidate-criteria(criteria), repair-window-criteria(criteria))
End long-range-plan

```

As can be seen the procedures are driven by two types of planning criterion:

- Candidate-criteria determine which visit to schedule next, such as:
  - o Plan high priority visits first

- Plan more constrained visits first
- Window criteria determine what window a visit should be assigned, such as:
  - Prefer plan windows that balance resource usage
  - Prefer longer plan windows (a plan window is up to 56 days in length)
  - Prefer windows early in the planning interval
  - Prefer plan windows early in the visits constraint window (i.e. provide contingency)

The system allows the initial guess and repair steps to utilize different criteria. Criteria are user activated and weighted. The impacts of different criteria are combined using a weighted average. In addition to the above built-in criteria the system supports the creation of mission specific criteria requiring a developer to implement object-oriented protocol methods. For example, the current instance of the WFIRST long range planner used priority based candidate criteria for some runs to ensure that core science programs got planned before GO programs. All runs used the generic resource leveling criteria to drive the selection of plan windows. This criterion both attracts visits to intervals with under subscription and pushes away visits to intervals with over-subscription. Parameters controlling the acceptable levels of under-subscription and the punishment for over-subscription were adjusted for the WFIRST mission. It is expected that more sophisticated mission specific criteria will be developed in future studies.

The following procedure is used in both the initial guess and repair steps. This routine takes a visit and a set of criteria and determines the best window for the visit by looping over the planning times and selecting the best window with respect to a set of criteria. The time looping is done such that no window that is strictly a subset of another window will be considered. This implicitly gives priority to larger windows. When marching toward the end of a constraint window the code will stop considering windows shorter than the nominal plan window size (56 days currently). Also note that the planning start and end interval over which windows are examined can be specialized for the visit at hand.

**Procedure find-best-window(visit, window-criteria)**

```

  Let best-candidate-value = 0
  Let best-candidate-window = nil
  Loop over time from planning-start(visit) to planning-end(visit)
    Let candidate-window = create a window for the visit at that time
    Let curr-value = evaluate(visit, candidate-window, window-criteria)
    If curr-value > best-candidate-value
      Best-candidate-window = candidate-window
      Best-candidate-value = curr-value
  End loop
  If (best-candidate-window != nil)
    assign-window(visit,best-candidate-window)
End find-best-window

```

The procedure `lrp-initial-guess` takes a set of visits and plans them with respect to two types of criteria. `Guess-candidate-criteria` control the order in which visits are selected for planning while `guess-window-criteria` help select which of the examined windows is

best for scheduling. This procedure loops through the visits once calling find-best-window to assign each visit.

**Procedure initial-guess**(visits, guess-candidate-criteria, guess-window-criteria)

```
    Loop while all visits have not been planned
        Use candidate criteria to select the next visit to plan
        find-best-window(visit-to-plan, window-criteria)
    End Loop
End initial-guess
```

The lrp-repair procedure is designed to remove resource oversubscription from the current schedule. The procedure works by finding a candidate visit that is scheduled in an oversubscribed area and then finds a series of swaps that will repair the visit. The system incrementally considers swaps with more intermediate visits. To move a visit V1 planned at an over subscribed time:

- At swap-level 0, the procedure find undersubscribed times where v1 can be directly planned.
- At swap-level 1, the procedure finds another visit V2 which can be directly moved to an undersubscribed region so that v1 can be planned without resource oversubscription during the time v2 was originally planned.
- At swap level 2, the procedure finds two additional visits v2 and v3 such that v3 can be directly moved to a resource undersubscribed time, v2 can be planned without resource oversubscription in v3s original window and finally v1 can be planned without recourse oversubscription during the time v2 was originally planned.
- Theoretically the code can work for any level of swaps. In practice level 2 is the highest that has been used.

Note that the procedure uses the find-best-window to actually move the visit. Different candidate selection and window criteria are used to help identify visits that are good for swapping and ensuring visits do not just get planned back in the same spot they used to be in.

**Procedure repair**(visits, repair-candidate-criteria, repair-window-criteria, swap-levels, iterations)

```
    Let level = 0
    While resources-oversubscription() and level ≤ swap-levels
        Let iteration = 0
        While resources-oversubscription() and iteration < iterations
            Use repair-candidate-criteria to determine a potential swap that
            moves a visit from an oversubscribed window to an under-
            subscribed region with swap-level number of intermediate moves
            For each visit in the swap
                find-best-window(visit, repair-window-criteria)
            iteration++
        End while
        level++
    End while
End repair
```

The SPIKE system also has infrastructure to support the maintenance of a plan as visits are executed over the course of a planning cycle.

### 3.3.3 Short Term Scheduling Engine

The WFIRST short term scheduling engine is driven by the long range plans created by the planner. In the short term engine each visit is limited to schedule in the plan window given by the input plan. As described above this two phase planning and scheduling system is required to allow for efficient execution.

Like the long range planning engine the SPIKE short term scheduling system provides many different scheduling algorithms that can be specialized and scripted. Before describing these algorithms some of the terminology used in the short term scheduler is defined. The short term scheduler divides time into fixed sized bins defined by an input time quantum. Visit durations are rounded up to the nearest quantum size to fit in the bins. So with a quantum of 5 minutes a visit requiring 8 minutes would require two units of quantum time to schedule and would result in two minutes of quantum loss. The experiments described below use a five minutes quantum. The schedule loss due to round off error is given in the experimental section.

The short term scheduler works to find schedules which have no conflicts for visits. For each possible quantum time that a visit can schedule the scheduling engine records the number of conflicts on that value. Conflicts occur due to another visit being scheduled on that time, or due to slews between an already scheduled visit, or due to timing links between visits (note that we have not yet input timing links in our visit set). For each visit the scheduler knows the conflict free values and the count of number of conflicts for conflicted values. Also the system as a whole tracks the minimum and maximum conflict values for all visits and quantum values. Initial runs of the planning system had long runtimes due to the conflict detection mechanism. On scheduling a visit the system determined if conflicts occur for all other visits. A runtime improvement was put in place such that potential conflicting visits can easily be detected. The improvement was driven by the observation that visits with non-overlapping plan windows cannot conflict with each other.

The inter-visit slew model in the short term scheduler takes into account quantum round off in that the slew overhead between targets can slide into the quantum round off error for a visit. For example, suppose a visit, V1, is 8 minutes long and the quantum is 5 minutes. Next suppose that the slew time from V1 to V2 is 1 minute. This one minute can occur during the two minutes of quantum round off due to the duration of V1. Consider a second example where the slew time between V1 and V3 is 6 minutes. In this case 2 minutes of the slew can be pushed into the quantum round off error of visit V1 but the slew will consume 4 minutes of the next quantum resulting in a one minute quantum loss due to the slew. All types of quantum loss are tracked in schedule statistics.

### 3.3.4 SPIKE Scheduling Steps

For WFIRST scheduling the following built-in early-schedule procedure is used extensively:

```
Procedure early-schedule(visits-to-schedule)
  Reset-scheduled-time(visits-to-schedule)
  // Schedule all visits early
  Loop until all visits-to-schedule are scheduled:
    Find the visit which can schedule at the earliest time with the
    minimum number of conflicts possible (resolve ties randomly).
    A simple variant resolves ties by selecting the longest visit (if
    there are multiple longest visits ties are resolved randomly).
    Schedule the visit at the early time
  // Resolve assignment conflicts
  Loop until no conflicts on scheduled times or until twice the length of visits-to-
  schedule
    Find a visit with maximum conflicts (resolve ties randomly)
    Assign the visit a time with minimum conflicts
  // Unschedule visits to remove conflicts
  Loop until no conflicts
    Find and unschedule a visit scheduled to a maximum conflict value
  // See if an unassigned visit can schedule in a conflict free time
  Loop through all unassigned visits
    Find a visit which has a value with no conflicts
    Assign the visit the earliest conflict free time
End early-schedule
```

The early-schedule procedure is looped in the run-early-schedules procedure below to run multiple iterations trying to find a schedule which minimizes the unscheduled-time() function which is the sum of the time of unscheduled visits.

```
Function run-early-schedules(iterations, visits-to-schedule)
  Let best-unscheduled-time = plus-infinity
  Let best-schedule = nil
  Loop iterations
    early-schedule(visits-to-schedule)
    If (unscheduled-time() < best-unscheduled-time)
      Set best-schedule = current-schedule()
      Set best-unscheduled-time = unscheduled-time()
  End Loop
  Return Best-schedule
```

After the scheduler run is complete the following metrics are collected:

- The number of visits that could not be scheduled without conflicts
- The duration of the visits that could not be scheduled (both in quanta and real time)
- The total slew time (both in quanta and in real time)
- The total time lost to quantum round off effects



- The schedule gap time (i.e. time with no activities including slews)
- The schedule non-science time (i.e. time with no visits)

In the metrics collected above the primary metric is the duration of the visits that cannot be scheduled without a conflict. Procedures were developed that take advantage of the tight coupling between long range planning and short term scheduling in the WFIRST application. A visit in the short term engine can only schedule within the plan window produced by the long range planner. Examining initial schedules it was found that some visits would schedule if the plan window was moved. Software procedures were defined which allow the plan window to move so that it overlaps gaps in the schedule. These procedures work like the first two iterations of the LRP repair step:

Procedure **adjust-plan-to-fill-gaps**(unscheduled-visits)

Loop over all unscheduled visits

Level 0: Find unscheduled visits that would fit in a schedule gap if the plan window is changed.

Loop over all unscheduled visits

Level 1: Find a scheduled visit that can move to a schedule gap (either with or without changing the plan window) such that an unscheduled visit can be scheduled in the moved visits place (either with or without changing the plan window)

End adjust-plan-to-fill-gaps

A common routine used in the experiments is to pair the procedures run-early-schedules() with adjust-plan-to-fill-gaps().

Procedure **schedule-early-and-fill-gaps**(iterations, visits)

run-early-schedules(iterations, visits)

adjust-plan-to-fill-gaps(remove-if-scheduled(visits))

End schedule-early-and-fill-gaps

The LRP and short term scheduling routines described were scripted together in different ways to run experiments as described in the experimental section below.

## 4 Scheduling Studies Results

### 4.1 Scheduling Study Input Parameters

The scheduling study used the observations in Table 2:

- 11384 visits with 1744.29 days of observing time
- 11176 schedulable visits with 1676 days of observing time
- 208 unschedulable visits with 69.8 days of observing time
- 374.4 days blocked out for microlensing

The scheduling studies utilized a six year or 2192 day time span from 10/31/2024 to 11/1/2030 and scheduled visits at a five minute time quantum. Table 5 gives information

on visits which were unschedulable in the planning system as they have no suitable constraint windows.

Program	Days Unschedulable	Number of visits
HLS	8.35	168
Coronagraphy	23.14	28
GO	38.32	12

**Table 5: Unschedulable Visits in the Strawperson Program**

We are in the process of analyzing each visit to find out why it is not schedulable.

## 4.2 Experiments Performed

While the WFIRST scheduler has been under development over the past year, it has been able to create schedules for many months and initial versions of the scheduler were run on initial versions of the strawperson observing program. These initial experiments demonstrated interesting results:

- The need to divide the problem into long range planning and short term scheduling.
  - o Initial runs just using a short-term scheduler were not able to complete due to performance reasons
- The need for additional runtime improvements such as speeding up the conflict detection routines in the short term scheduler
- The use of a resource criteria in long range planning is critical to obtaining efficient short term schedules
  - o Initial runs showed that a resource driven long range plan reduced the number of unscheduled visits from 25% to less than a percent for an early set of programs.

The results of these initial experiments informed the implementation and the parameters used in the experiments reported below using the more complete strawperson program.

A series of three scheduling runs were performed using the complete strawperson program where each run builds on the next by engaging additional software mechanisms or extra processing steps. While the three schedule runs only scratch the surface of possible experiments, they do demonstrate that the scheduler works and can find interesting features of the scheduling space. The first run created an LRP using just the resource criteria and then ran a short-term schedule over all visits. Using the above procedures this is scripted as:

- long-range-plan(visits, { resource-criteria })
- early-schedule-and-fill-gaps(visits, 16)

Analysis of this run found that while the resulting schedule had high spacecraft efficiency, most of the visits that did not schedule were from the core science programs. Virtually all the GO observation visits were scheduled while many of the core science visits were not

scheduled. A second scheduling run attempted to fix this problem by giving priority in short term scheduling to the core programs. This run can be scripted as:

- long-range-plan(visits, { resource-criteria })
- Let core-visits = remove-if-not-core(all-visits) //Schedule the core science first
- Early-schedule-and-fill-gaps(core-visits, 16)
- Let go-visits = remove-if-not-go(all-visits) // Schedule GO science second
- Early-schedule-and-fill-gaps(core-visits, 16)

Analysis of this run found that this approach significantly increased scheduling of core science programs at the cost of a lower overall spacecraft utilization. However, on more detailed examination the schedule was found to have many GO programs scheduled in the first few years of the six year mission. This is in contrast to the mission goal of scheduling the core science first. Analysis showed that the LRP used in these runs was biased to plan GO programs early in the 6 year period. This LRP had no candidate selection criteria. With no candidate selection criteria the planner defaults to the lexical order of visit IDs. As most of the GO programs have lower proposal numbers they were scheduled first. The planner also has an early bias. The first time which maximizes the resource criteria will be selected. As a result the long range plan biases the GO programs early in the six year planning window. In order to fix this a candidate selection criteria was defined which plans the core science programs first. This run can be scripted as:

- long-range-plan(visits, { resource-criteria, plan-core-first-criteria })
- Let core-visits = remove-if-not-core(all-visits) // Schedule the core science first
- Early-schedule-and-fill-gaps(core-visits, 16)
- Let go-visits = remove-if-not-go(all-visits) // Schedule GO science second
- Early-schedule-and-fill-gaps(core-visits, 16)

This run resulted in a modest decline in space craft efficiency and much of the core science being executed earlier in the six year period.

Detailed results are now presented for each of these three runs. The runs are summarized and named below to help in the presentation:

- **Max Spacecraft Utilization** - Maximize science utilization independent of which programs are scheduled.
- **Max Core Science** - Maximize the scheduling of the core science programs (i.e. everything but the GO program) and then space craft science utilization
- **Early Max Core Science** – Same as above but prefer to schedule core science early

While detailed runtimes were not collected for these runs each took about 3-5 hours to run the process from long range plan creation through short term scheduling. This was done on an older laptop with 8 gigabytes of memory.

Metrics collected from the three schedule runs are shown in table 6 below. Some highlights of the results:

- The highest spacecraft efficiency and the lowest amount of unscheduled science results from scheduling without regard to core science versus GO programs as done in the Max-Spacecraft-Utilization run.

- Scheduling without regard to core science versus GO programs results in all the GO programs being scheduled and all the unscheduled visits coming from the core programs.
- Scheduling approaches which prefer core science result in more core science being scheduled at the expense of lower overall spacecraft utilization and slightly higher slew durations.
  - o These results suggest new studies to determine:
    - The cause of the extra slew overhead when more core science is scheduled
    - Whether the core science programs cannot be scheduled together due to inherent resource conflicts
    - Note that these two studies conflict in that visits which require the same time for scheduling should have relatively near targets.
- Scheduling core science early resulted in significantly more of the core science being dropped from the schedule. Evidence that core science was moved early will be shown graphically.
- The loss due to scheduling with fixed 5 minute long time bins is less than 1% of the six year schedule time span.
- Overall the system does well with respect to getting close to the maximum possible science efficiency (i.e. when all 1674.49 days of potential science are scheduled).

Some additional analysis was done on the unschedulable visits and it was found that there is a strong correlation between being unscheduled and the length of an observation. For example in the Max-Core-Science run the 6.42 days of unschedulable core science came from 7 visits. Also the 83.35 days of unschedulable GO programs came from 68 visits yielding an average of 1.25 days per visit. In contrast the average duration of the 4829 visits in the GO pool is 0.11 days and the average duration of scheduled GO visits is 0.09 days. A final experiment was done and confirmed that all but two of the unscheduled GO visits have schedulable intervals during gaps in the schedule. These are intervals that are too small to fit the day long visits but big enough to schedule some of the visits. These results strongly suggest that making smaller more flexible GO and core science observations will improve overall science efficiency.

Schedule plots were created which show how each day in the six year period is allocated to the core science programs, GO visits, slew time, and gap time. Each plot consists of a row per year where the visits, slews, gap time scheduled per day are given by a color coding in successive columns. The color code key for all plots is given in figure 7. The plots for the three runs are given in Figures 8-10. Some observations:

- From all three plots it can be seen that days with mixed programs or GO programs have higher slew costs.
  - o This makes sense as GO programs have different targets and typically the core science programs view different parts of the sky.
- The increase in core science observations from the max-spacecraft-utilization run to the max-core-science can be seen at the very start of the plot and in the winter of 2029.
- The plot for the Early-Max-Core-Science clearly shows that core science is moved forward at the expense of GO observations.

	<b>Max Spacecraft Utilization</b>	<b>Max Core Science</b>	<b>Early Max Core Science</b>
Unscheduled Duration in days	51.77	89.77	97.75
Percentage unscheduled	3%	5%	6%
Unscheduled duration from core science programs in days	51.77	6.42	23.84
Unscheduled duration from GO programs in days	0	83.35	73.91
Total scheduled duration in days	1622.72	1584.72	1576.74
Total Slew Duration in days	55.41	62.06	64.52
Total quantum loss in days	14.31	15.46	15.62
Science Efficiency	91.1%	89.37%	89.0%
Maximum Possible Science Efficiency	93.5%	93.5%	93.5%
Spacecraft Efficiency	93.6%	92.2%	91.9%

**Table 6: Metrics for schedule runs.** Science efficiency is the percentage of the 6 year planning interval spent executing visits. Maximum science efficiency is the highest possible science efficiency given the duration of schedulable visits in the strawperson program. Spacecraft efficiency is the percentage of the 6 year planning interval spent either executing visits or performing slews between visits. Note that visit durations include slews between exposures and other overheads. The system is not currently instrumented to measure the exposing efficiency which can be defined as the percentage of the 6 year planning interval spent taking exposures.

While these experiments are in no way exhaustive they do demonstrate that:

- The planning and scheduling system is capable of creating efficient WFIRST schedules in reasonable runtimes (3-5 hours).
- The planning and scheduling mechanisms available in the SPIKE system can be used to impact scheduling results to achieve desired mission goals as demonstrated by the ability to prefer to schedule core WFIRST programs early in the six year cycle.

It should be noted that the SPIKE system parameters and subroutines do not provide the only means to affect the scheduling outcome. It is also possible (common, in fact) to add

constraints on individual visits, which force them to, e.g., execute within a given absolute or relative time frame, in a particular sequence, or a manner that is uninterrupted by other types of activities. One might imagine, e.g., that science drivers could exist to execute the HLS in large uninterrupted blocks, or in a particular pattern (like vacuuming the floor, instead of picking tiles in "random" order). Or, there might emerge high-level WFIRST Project decisions on the order in which the science or survey components should execute, or the fraction of time to devote to them in the first year. All such constraints can be flexibly incorporated into the existing scheduling framework presented here.



**Figure 7: Key for scheduling plots given in Figures 8-10**

Time used by each program in schedule (counted in seconds)

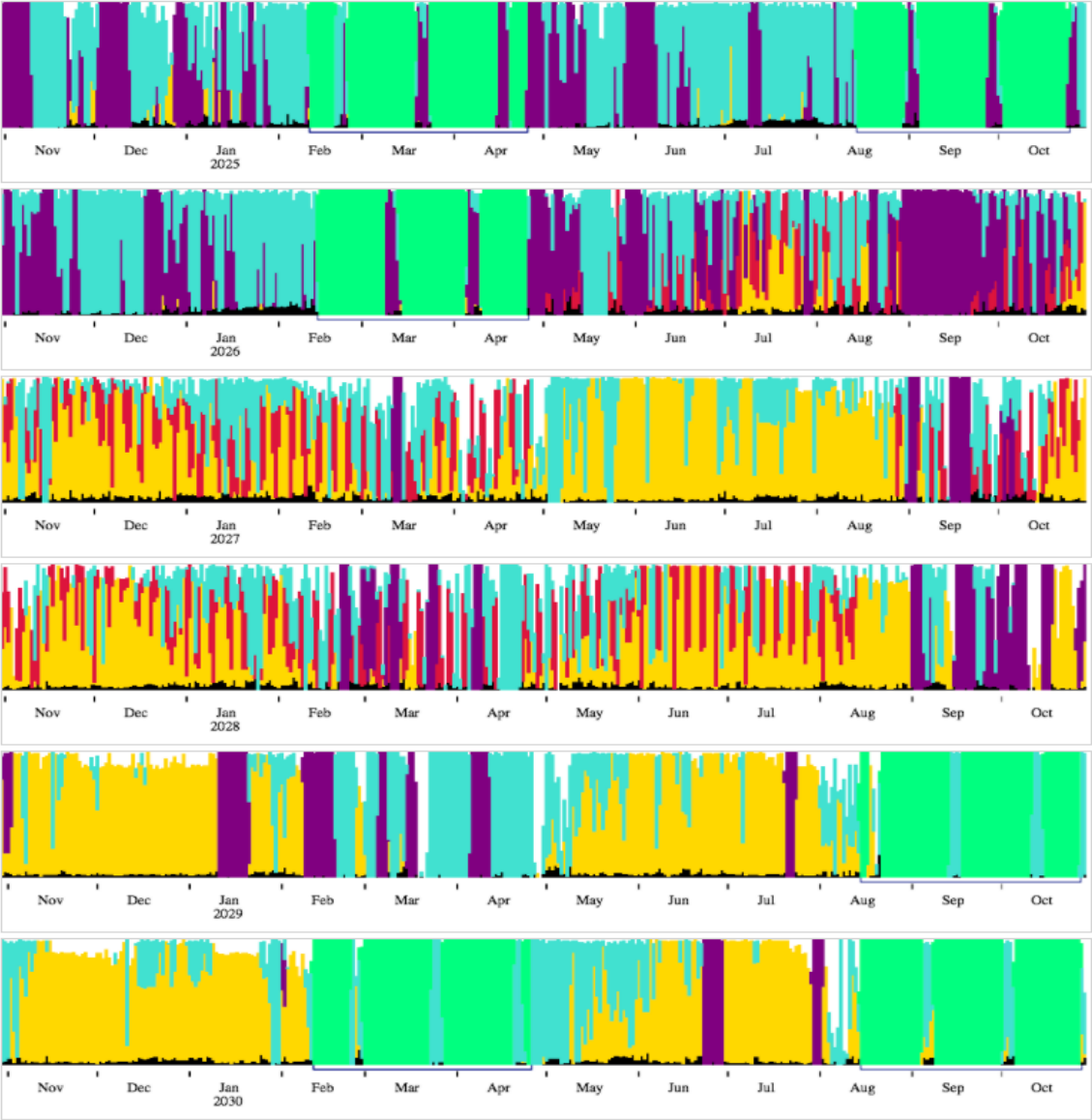
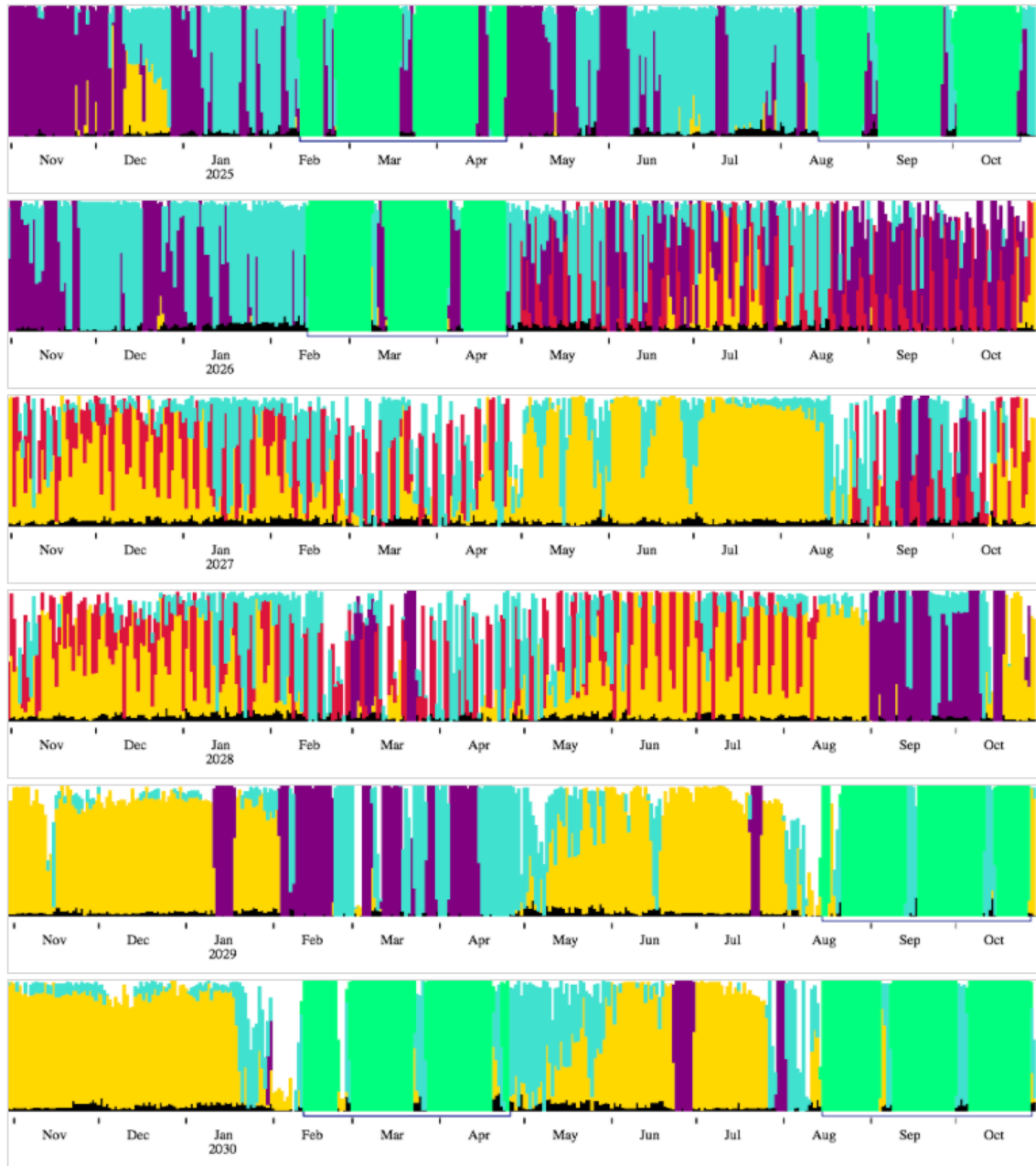


Figure 8: Max Spacecraft Utilization – Schedule Plot

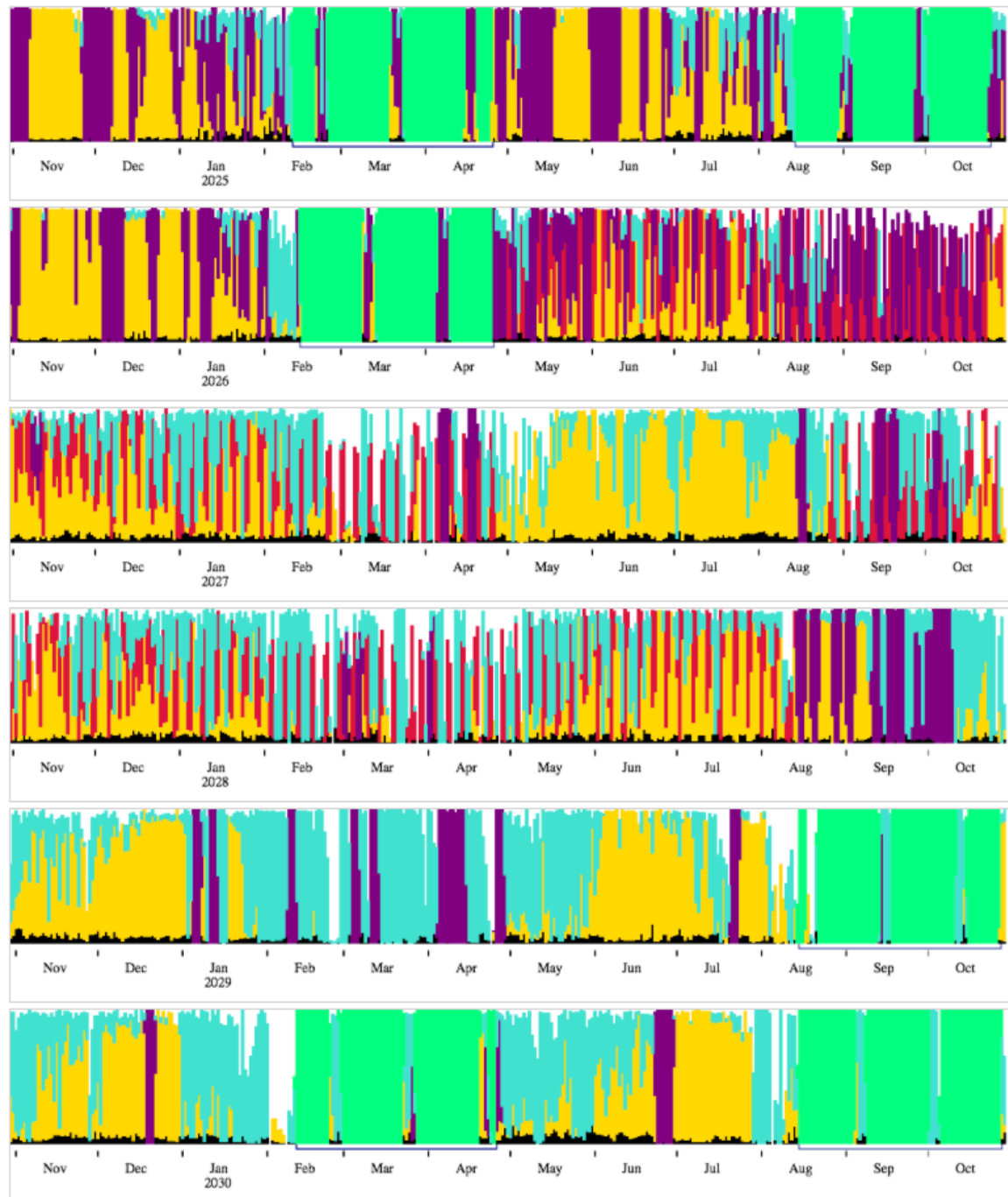
**Time used by each program in schedule (counted in seconds)**



**Figure 9: Max Core Science Schedule Plot**



**Time used by each program in schedule (counted in seconds)**



**Figure 10: Early Max Core Science Schedule Plot**

## 5 Future Work

While the observing program and software systems, described above, represent most core mission features, each is an initial version and is missing components. Significant areas of improvement will be required to reach final conclusions about the feasibility of scheduling the WFIRST science program. This future work is outlined below:

Improvements to the Strawperson observing program:

- Ensuring all visits in the program are schedulable
- Breaking visits into smaller pieces to allow for scheduling flexibility
- Changes to targets as inherent scheduling conflicts are determined.

Additions to the scheduling constraint model:

- Timing links between observations (supported in SPIKE planning and scheduling infrastructure)
- Modeling roll links between observations (e.g. orient V1 from 10-20 degrees from V2)
- Assigning all observations a scheduled roll
  - o While modeling roll links and assignments is supported in SPIKE new models will be needed to handle scheduling data at a five minute interval over six years
- Guide stars and Zodiacal light

Additional Scheduling studies:

- Studies at L2 orbit – The constraint calculation routines support this
- Tools and analysis to determine if unschedulable visits are due to inherent resource conflicts in the observing program or due to less than optimal scheduling algorithms
- Schedule visualization tools
- Additional scheduling preferences
  - o Prefer to schedule core science over GO programs
  - o Prefer to schedule core science programs early
- Multi-Objective Scheduling - The current scheduler is configured to optimize the amount of time spent observing (i.e. space craft efficiency). However, there are other objectives that could be measured:
  - For each of the core programs a minimization objective could measure the observing time for the program not scheduled
  - For each of the core programs a minimization objective could measure the extent to which the program is scheduled early
  - For GO programs we can maximize the number of programs that reach science completion. A schedule which schedules some of every GO program may not yield scientific results

A multi-objective scheduler would explore the search space producing a wide variety of schedules that explore the trade-offs between the different objectives. A multi-objective scheduler produce a pareto-surface of schedules where no schedule on the surface is worse with respect to all objectives than another schedule in the set. Visualization tools

can then be used to explore the trade-offs in the schedules between the different optimization objectives.

## Acknowledgment

This work was carried out under contract with the WFIRST Study Office at the NASA Goddard Space Flight Center (GSFC), as part of joint preformulation science center studies by the Space Telescope Science Institute (STScI) and the Infrared Processing and Analysis Center (IPAC).

## References

Giuliano, M.E. and Johnston, M.D. 2010. “Visualization Tools For Multi-Objective Algorithms”. Demonstration: *International Conference on Automated Planning and Scheduling (ICAPS)*, Toronto, Canada.

Hirata, C. 2015. “Notes on the WFIRST Observing Plan – Version 3”. Obtained from private exchange.

Johnston, M.D. and Miller, G. 1994. “Spike: Intelligent Scheduling of Hubble Space Telescope Observations”. *Intelligent Scheduling*, eds. M. Fox & M. Zweben (Morgan Kaufmann: San Mateo)

Spergel, D. *et al.* 2015. “Wide-Field InfraRed Survey Telescope-Astrophysics Focused Telescope Assets WFIRST-AFTA 2015 Report by the Science Definition Team (SDT) and WFIRST Study Office”. <http://arxiv.org/abs/1305.5422>

## APPENDIX A – Details on WFIRST Science Strawperson Programs

Strawperson programs were specified in EXCEL spreadsheet or comma-separated values text files. The specification format is as follows:

- HLS, SN and coronagraphy– Each line specifies (1) observation ID, (2) RA, (3) DEC, (4) Duration in seconds, (5) program type (3 for SN, 5 for HLS imaging, 6 for HLS spectroscopy, 7 for coronagraphy), (6) position angle, if any, (7) time interval to execute, if any.
- General observer – Most of the GO observations are to be done at multiple targets. Thus, a GO program is given in two files, a target file and an observation file. Observation file contains observation ID, information on where the observation comes from (ex. JWST SODRM program 9XXXXX), type of the observation (imaging, grism, IFU), the number of targets, size of mosaic, number of dither points to be done at each mosaic point, number of filters to be used at each dither point, exposure duration at each filter and dither point. It also contains extra scheduling constraints in text (ex. Repeat once per year for three years. Repeat using 3 position angles >15 degrees apart). Target file contains a list of RA and DEC for each observation.

These specification files are available upon request. We are in the process of converting these specification files to JSON files in an unified JSON schema.

### A1. High Latitude Survey

The High Latitude Survey (HLS) is described in detail in the WFIRST SDT report (Spergel et al. 2015), from which information relevant to this scheduling study has been obtained, as summarized here. The overall layout of the HLS is a contiguous region covering  $2227 \text{ deg}^2$  consisting of imaging in four near-IR filters (Y, J, H, and F184), covering the wavelength range  $0.92 - 2.00 \text{ } \mu\text{m}$  to limiting depths  $\sim 25.8 - 26.7 \text{ AB}$ . In addition, the same region is covered using grism observations ( $1.35 - 1.89 \text{ } \mu\text{m}$ ), to obtain complete spectroscopic information on the entire survey area.

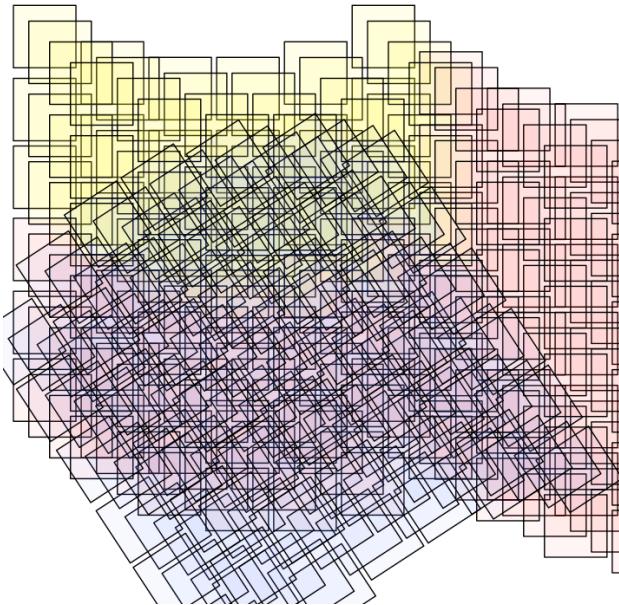
The survey observations are arranged in a series of “fields”, each corresponding to the full coverage area of the WFIRST imaging camera (approximately  $45'$  in width, with a total area of  $0.281 \text{ degree}^2$ s). For each field, 3 to 4 exposures are obtained in each of the different filters, with the exposures acquired using a small-step dither pattern with offsets of about  $\frac{1}{4} \text{ SCA}$ , horizontally and vertically simultaneously. Once this pattern is complete for each field, the observatory slews to the adjacent field and repeats the pattern, thereby acquiring a sequence of strips across the sky to cover the entire HLS survey area.

In addition, each field is covered in a second pass a half-integer number of years later, at which point the full depth coverage is achieved. The second pass observations are obtained at a different roll angle relative to the first pass, in order to provide observations of sources at all pairs of positions on the focal plane and provide constraints for the self-calibration solution. The non-integer pixel phase sampling resulting from this different roll angle also helps to provide Nyquist sampling of the PSF at J-band and longer

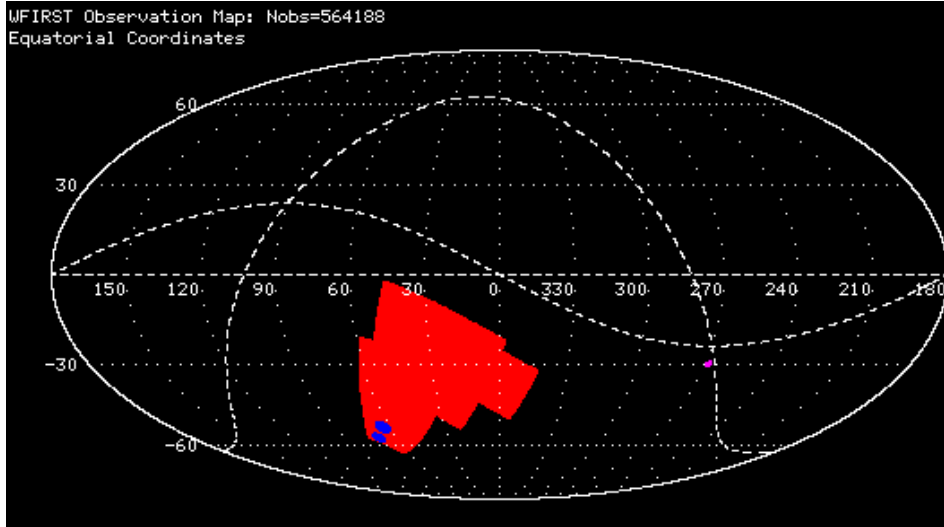
wavelengths, while the time frequency of these repeat observations are distributed throughout the mission in order to account for long-term drifts in the calibration.

The grism observations follow a similar structure to the imaging observations, with the main difference being that there are 4 passes (instead of 2), with 2 small-step dithered positions during each pass (instead of 4). The 4 passes are all obtained at different roll angles, which is important for the grism data calibration. Two of these passes are obtained in the “leading” geometry relative to Earth’s orbit, and the other two in the “trailing” geometry, so that both northward and southward dispersion directions are obtained. This removes systematic errors from the wavelength calibration, in particular the offset of emission line centroids relative to the continuum centroid for each source, which is used as a reference location when extracting the grism data.

For this scheduling study, the full set of exposures were grouped into “visits”, where a given visit generally consists of all the dithered exposures necessary to observe one field, in all filters, during a single pass. This structure enables efficient scheduling of exposures that are very close together on the sky, as the  $\frac{1}{4}$  SCA offsets are most effectively handled internally to each visit. Adjacent fields, and also second-pass observations, are then handled as separate visits. This structure enables the HLS observations to be scheduled throughout the mission, starting with the first-pass observations of each field, and followed a half-integer number of years later by the second-pass observations at a different roll (as well as the third and fourth pass observations at different rolls for the grism observations). The total time used is thus 2.01 years including overheads, and accounting also for the effects of field distortions on the tiling solution, and the slew and settle times to the next field (which are handled as separate visits).



**Figure 1** Example layout of the exposure pattern for each field in the HLS (courtesy of C. Hirata; see Spergel et al. 2015), where a single field is approximately  $45'$  in width. The yellow exposures indicate the 4-point dither pattern obtained during the first pass of the field, with offsets of  $\sim \frac{1}{4}$  SCA horizontally and vertically, for each dither position, in order to cover the chip gaps. The red exposures below and to the right indicate the same structure for adjacent fields, all for the first pass. Finally, the blue exposures indicate the structure for the second pass, obtained a half-integer number of years later at a different roll angle. These patterns are repeated across the entire  $2227 \text{ deg}^2$  area of the HLS.



**Figure 2** The total HLS footprint (courtesy of C. Hirata; see Spergel et al. 2015), where the red area indicates the 2227 deg<sup>2</sup> area of the full survey, while the blue regions indicate the supernova survey fields.

## A2. Supernova Search

The supernova search is focused on two locations within the HLS near the ecliptic pole, using the Wide Field Imager to monitor these regions for a total of  $\sim 6$  months integration over a 2-year period, using visits that are repeated every 5 days. The search would be carried out using two filters to initially discover the supernovae, with subsequent IFU observations to obtain spectral information about the supernova candidates and their host galaxies.

This survey consists of three tiers, corresponding to different area / depth / redshift regimes, as follows:

- Wide, shallow:  $z < 0.4$ , search area of 27.44 deg<sup>2</sup> (7 x 14 WFI fields in total)
- Medium:  $z < 0.8$ , search area of 8.96 deg<sup>2</sup> (4 x 8 WFI fields in total)
- Small, deep:  $z < 1.7$ , search area of 5.04 deg<sup>2</sup> (3 x 6 WFI fields in total)

On average, each visit consists of 8 hours for discovery imaging and a further 7 hours for lightcurves, while the deep IFU spectra and galaxy reference observations would be an additional 7 hours and 3 hours respectively.

For this scheduling study, the observations were grouped into a total of 288 visits (144 visits for each of the two pointing locations), spread throughout the 2-year observing season. During each 5-day window, there is some flexibility in how the visits can be scheduled, with the only constraint being that each group of visits be completed during each 5-day window.

### **A3. Coronagraphy**

The 2015 SDT report on WFIRST-AFTA does not explicitly present a complete design reference mission (DRM), which would lay out specifics as to exposure times, targets, and observing strategies. The report does detail potential programs that can be reasonably joined together into a notional main program. We based the program on a targeted high contrast imaging survey of planet host stars detected via the radial velocity (RV) method, the characterization of a subset of the RV planets, and a blind search around nearby stars.

The SDT report (Table F-5) shows a notional DRM and gives some estimates for overheads. Exposure times are also listed for RV targets in Table (2-7) of the SDT report. The overheads are ~6 hrs for a typical discovery image, but no details are given for estimating overheads associated with spectral characterization observations. For those programs it was assumed that these took 12 hours per band. Programs were broken up into "discovery" (integration time+6 hours per filter), "characterization" (3 days+0.5 day for overhead per band). Exposure times for the blind search program assumed a similar observing strategy to the RV discovery program. Exposure times and targets for the blind search were kindly provided by D. Savransky and information was derived from the 0.8 mas jitter 30x post-processing case.

The final program was designed as follows. The RV "discovery" program consisted of 14 RV systems imaged in 4 Filters for a total of 1440 hrs vs. 960 assumed in the SDT and 14 days of overhead). The RV "Characterization" consisted of spectra of 8 RV planets in 4 bands each (here we assume that 10 of the days here would actually be taken up with Disk spectral characterization rather than planets). The total integration time is then 2304 hrs vs 2592 assumed in SDT, along with 16 days of overhead. The Blind "discovery" program used 41 targets. The exposure times used were divided by 4 and each target was visited 4 separate times to simulate multiple visits over a longer period of time. This corresponds to 3252 hrs vs 1680 assumed in the SDT, with 41 days of overhead. In total, the SDT assumed 56 days of total overhead, where with our assumptions the total was 71 days. The total time for the main mission was 363 days.

### **A4. General Observer program**

This section describes the methodology to develop a set of notional GO observations for WFIRST-AFTA to aid in executing a realistic scheduling efficiency study.

The primary goal has been to provide a mix of observations that cover a variety of instrument modes, coordinates, and observing strategies in a manner timely enough to be useful for the scheduling study, rather than direct relation to the actual science goals a rigorous GO science program might take. For the sake of efficiency, therefore, the programs have generally been adapted and modified from other sources with varying degrees of concern for how well they represent sensible WFIRST observations. Care should be taken when using these mock GO programs for other purposes.

The programs in this study primarily focus on the imaging, coronagraphic, and, to a lesser extent, grism modes. There are a limited number of IFU observations, but it is assumed that this will not strongly affect the outcome of the scheduling study. In general, no attempt has been made to strike a realistic balance of observing time between the various potential modes of the WFI and the CGI.

The primary sources on which the mock GO programs are based are the JWST Science Operations Design Reference Mission (SODRM, <http://www.stsci.edu/jwst/science/sodrm/SODRM-Revision-C.pdf>) and the WFIRST science 1-pagers included in the SDT 2015 report. The 1-pagers tend by nature to better represent actual WFIRST science, although the SODRM programs have been useful for increasing the overall number of programs and developing possible grism programs. Note that the 1-pagers were largely written before the current mission parameters were adopted.

The total observing time across all programs is roughly 7 months, without overheads. The programs are summarized as follows:

**NIRCAM SODRM** — These are programs based on the JWST NIRCAM SODRM. This was the first set of programs developed for the scheduling study, and therefore tend to be the most primitive with respect to how well they represent WFIRST science. They do cover a range of science topics, sky positions, and integrations times, however. In order to avoid excessively long programs the WFIRST integration times were generally kept the same as for JWST. The larger field of WFIRST was considered for tiled JWST observations, however. Targets were obtained directly from APT using the SODRM program names. Coronagraphic programs were removed and modified, which is described later. The total exposure time in this group is 1358 hours.

**WFIRST SDT 1-pagers** — These are based on the WFIRST science 1-pagers collected in the 2015 SDT report. Like the SODRM, they cover a range of science topics, although in general these documents outline the science in broad terms and only provide limited information on what the specific observations would actually look like. Some license was therefore taken to identify reasonable targets, filters, and other parameters. The programs are again mainly imaging, though there are two that use the IFU. Most of the programs tend to be less than 100 hours, although some are up to 300 hours. In some cases these programs are for follow-up of objects discovered in the HLS. A large proportion of these programs have targets that were chosen at random within some specification, for example, to be at high galactic latitude or within the HLS footprint. Total exposure time in this group is 2120 hours.

**GO proper motions** — These cover the science 1-pagers that dealt with proper motions. In general they have long cadence times, with two or three visits separated by one to three years, although one program uses a short cadence (1 hour) to look for Kuiper belt objects. The former category includes a relatively large (240 hr) Galactic plane survey that is meant to represent the science from multiple 1-pagers. The programs in this group replace entries in the previous list. Some input was obtained from 1-pager authors, notably Jason Kalirai and David Ardila. This group contains some unusual coordinate restrictions; for example, PM06 uses coordinates arranged along several great circles in



order to represent Milky Way stellar streams. Total exposure time in this group is 880 hours.

GO grism — This is a set of grism observations derived from the JWST NIRSPEC (particularly MSA) and NIRISS SODRM. Although they are based on the SODRM parameters and targets, more effort (compared to those derived from the NIRCAM SODRM) was put into framing programs that were somewhat better suited to WFIRST. Again, however, the WFIRST observations will ultimately be quite different from their JWST counterparts. Total exposure time in this group = 860 hours.

GO Coronagraphy — Unlike the other programs, these were inspired by a combination of the 2015 SDT report for the main science mission and by NIRCAM SODRM programs that relied on the coronagraphic observations. This reflects the probable desire of the community to pursue science goals separate from the main CGI science program (as encoded in the NIRCAM programs, which are short and assume that full contrast may not be achieved) in addition to targets that may be pursued with the nominal CGI observing setups as envisioned in the SDT report. The programs for GO coronagraphy are then divided into three main groups—“discovery”, “characterization”, and “snapshot”. Discovery and characterization observations followed similar assumptions as to the main CGI notional coronagraphic program described in Section A4 above. Seven stars with directly imaged planets and the star Tau Ceti were chosen for the GO discovery. The characterization group included four additional RV planets drawn from the SDT CGI RV planet sample. The snapshot sample consisted of 6 extragalactic targets, 12 previously resolved debris disk stars, and 18 protoplanetary disk stars. These targets were all drawn from NIRCAM SODRM programs. A constant exposure time of 3.3 hours was assumed for these snapshot programs. The total time for this program is 2064 hours.

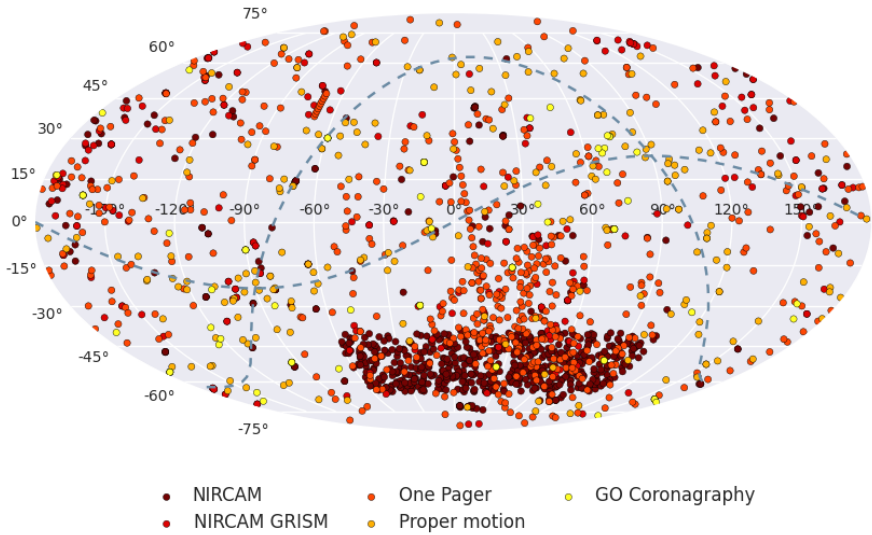
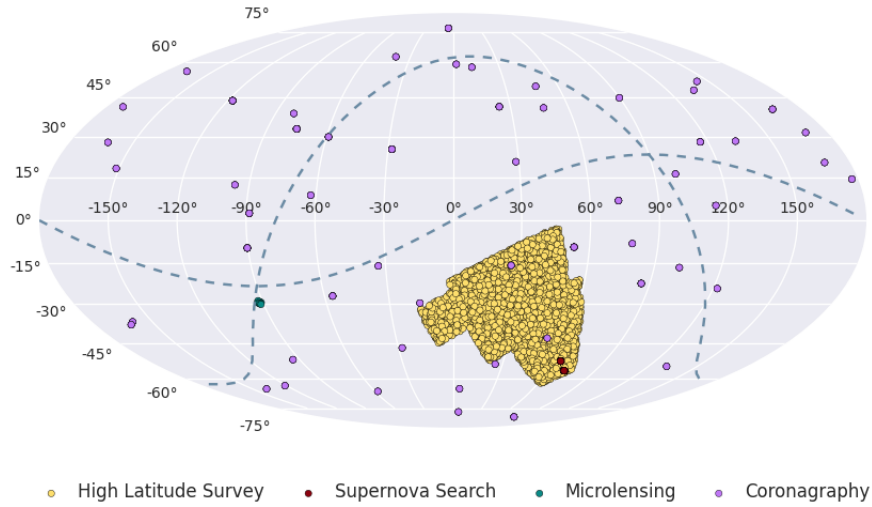
There are also some additional notes generally on how the programs were generated. For most of the GO programs, overheads are not included. The exception to this are GO coronagraphic discovery and characterization programs that use the same overheads as for the main CGI mission. For simplicity, each target in a non-coronagraphic program uses the same combination of observing parameters, including exposure times. In some cases where different modes or exposure times were clearly needed, a single SODRM or 1-pager program was divided into multiple GO programs. For example, a tiered approach was adopted for grism observations of extragalactic deep fields. For grism observations, a baseline of 3 roll angles per target was used, as this will be useful for disentangling sources<sup>1</sup>. Roll angles should be separated by >15 degrees. A nominal Group Within parameter of 53 days was adopted from the JWST SODRM for non-coronagraphic programs, but this is not well justified in most cases. For most cases the ordering and timing of observations is not critical. The exceptions to this are the proper motion studies, where the cadence was noted. There are also a small number of monitoring programs that require constant monitoring of a single target for a given period of time, e.g., grism observations of transiting exoplanets.

---

<sup>1</sup> Casertano, Stefano *et al.* 2015. “Slitless Slitless Grism Spectroscopy with WFIRST: Observing Modes and Strategies”. WFIRST-STScI-TR1506.

## A5. Target Plots

The top plot shows the target locations of the strawperson programs for the core science for WFIRST, with high latitude survey (yellow), supernova (red), microlensing (blue) and main coronagraphy (light purple). The dotted lines are the ecliptic equator and the galactic equator. The bottom plot shows the target locations of the general observer strawperson programs.

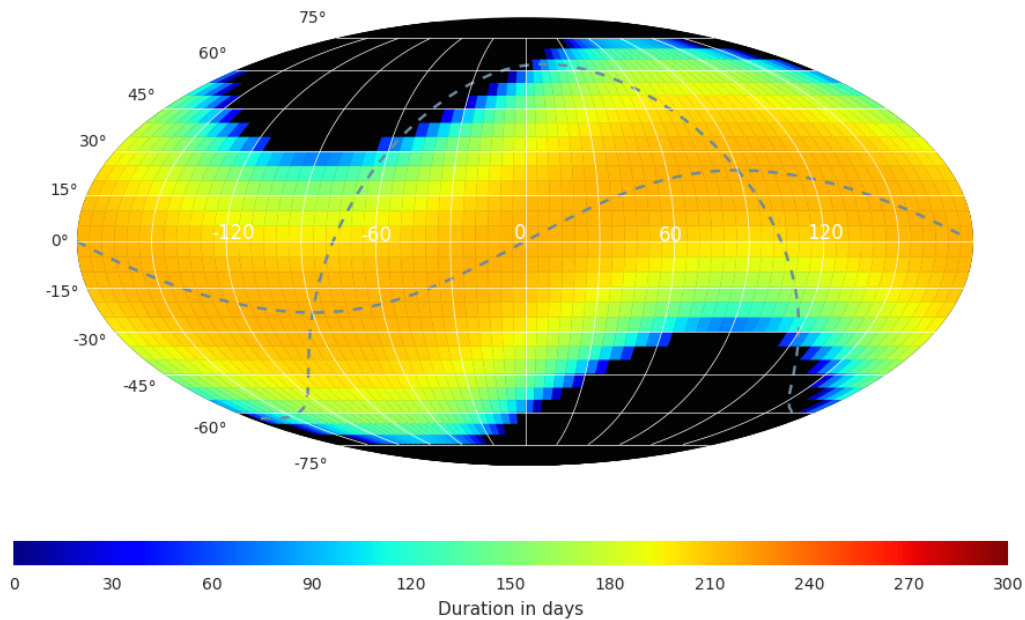


## APPENDIX B – Effects on Target Schedulability due to the Sun, Earth and Moon Constraints with a Geosynchronous Orbit

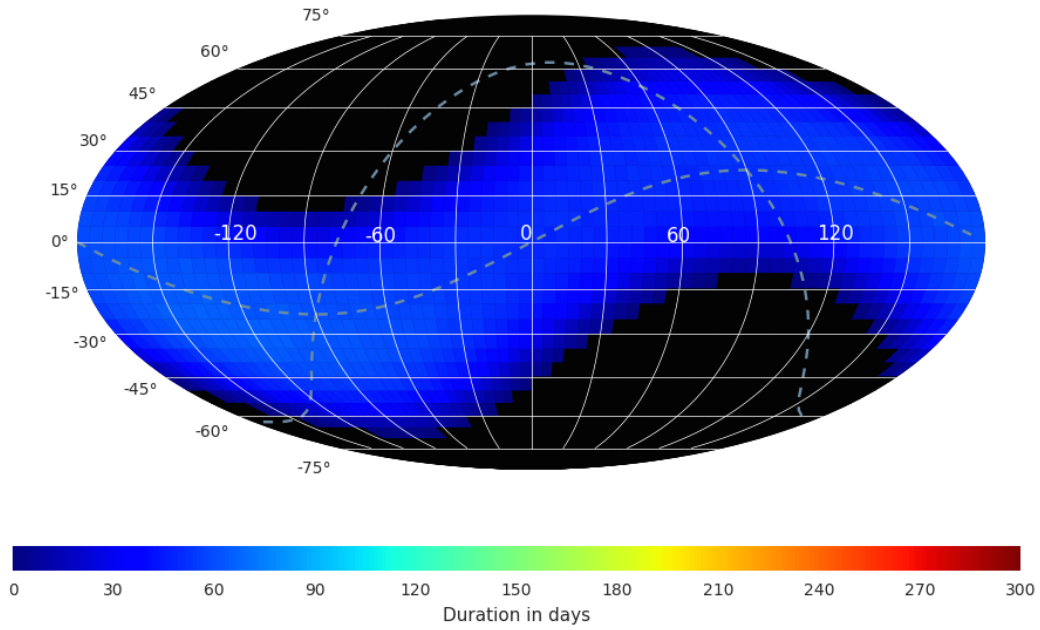
**B1. Total length of time in days affected by each constraint in year 1** (from 10/31/2024 to 11/1/2025).

Each plot shows the total length of time in the year the target cannot be observed due to the angle limit between the telescope line of sight and the object.

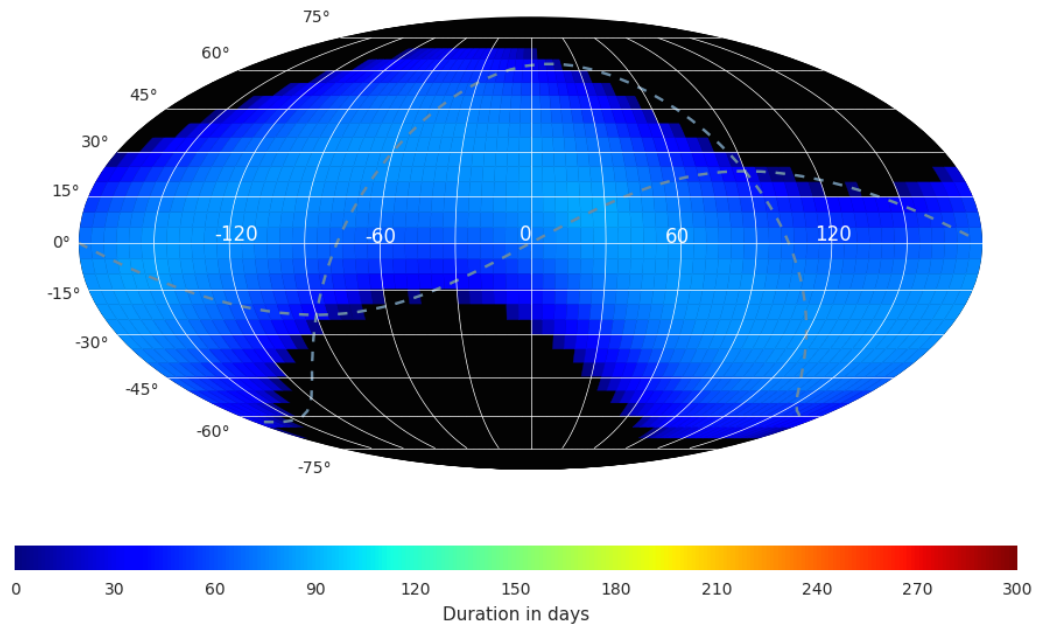
**Sun** – The constraint enforces the angle to be between 54 and 126 degrees. The maximum value in the plot is ~217 days.



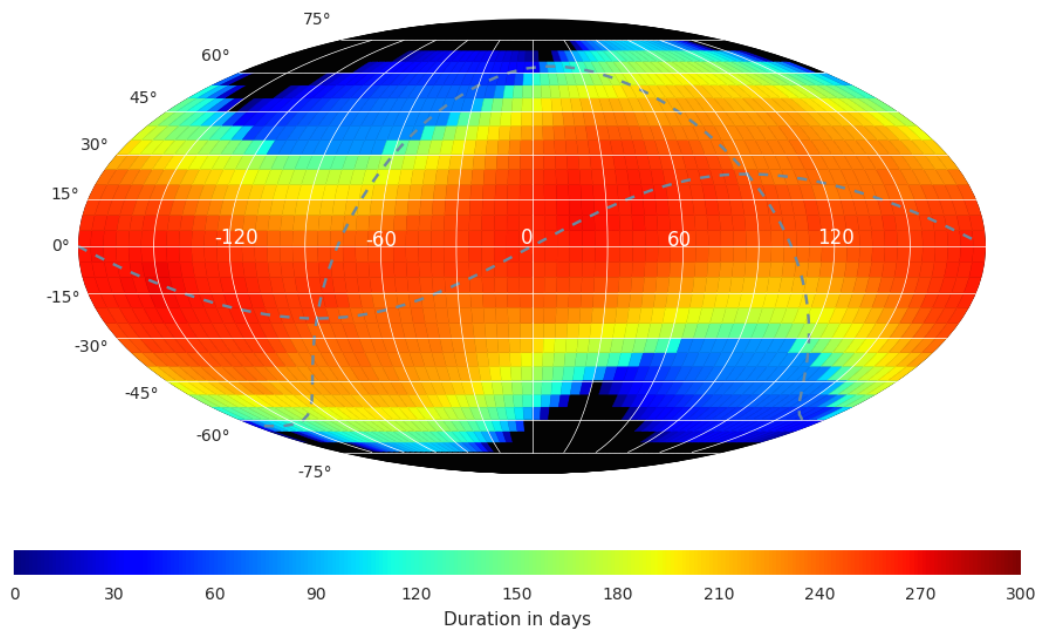
**Moon** – The constraint enforces the angle to the limbs be at least 35 degrees. The maximum value in the plot is ~66 days.



**Earth** – The constraint enforces the angle to the limbs be at least 35 degrees. The maximum value in the plot is ~88 days.



**Sun, Earth and Moon constraints combined** – The maximum value in the plot is ~271 days.



**B2. Regions where each constraint has effects on**

