



Automated Scheduling of Doppler Exoplanet Observations at Keck Observatory

Luke B. Handley¹ , Erik A. Petigura¹ , Velibor V. Mišić², Jack Lubin¹ , and Howard Isaacson^{3,4} ¹Department of Physics & Astronomy, University of California Los Angeles, Los Angeles, CA 90095, USA; lhandley@caltech.edu²Anderson School of Management, University of California Los Angeles, Los Angeles, CA 90095, USA³501 Campbell Hall, University of California at Berkeley, Berkeley, CA 94720, USA⁴Centre for Astrophysics, University of Southern Queensland, Toowoomba, QLD, Australia

Received 2023 November 17; revised 2024 January 12; accepted 2024 January 16; published 2024 February 21

Abstract

Precise Doppler studies of extrasolar planets require fine-grained control of observational cadence, i.e., the timing of and spacing between observations. We present a novel framework for scheduling a set of Doppler campaigns with different cadence requirements at the W. M. Keck Observatory. For a set of observing programs and allocated nights on an instrument, our software optimizes the timing and ordering of ~ 1000 observations within a given observing semester. We achieve a near-optimal solution in real-time using a hierarchical Integer Linear Programming framework. Our scheduling formulation optimizes over the roughly 10^{3000} possible orderings. A top level optimization finds the most regular sequence of allocated nights by which to observe each host star in the request catalog based on a frequency specified in the request. A second optimization scheme minimizes the slews and downtime of the instrument. We have assessed our algorithms performance with simulated data and with the real suite of Doppler observations of the California Planet Search in 2023.

Unified Astronomy Thesaurus concepts: [Astronomical methods \(1043\)](#); [Observational astronomy \(1145\)](#)

1. Introduction

The discovery and characterization of extrasolar planets with the Doppler technique requires careful attention to the observational “cadence,” defined as the timing of and spacing between observations of a given target. The census of known Doppler extrasolar planets have orbital periods ranging from only a few hours to several decades (Howard et al. 2012; Fulton et al. 2021). Thus, Doppler surveys must be tuned to appropriately sample the orbital periods of interest.

Moreover, time-variable surface features produce shifts in the stellar spectra that register as Doppler shifts, yet have nothing to do with the stars motion around the star-planet barycenter (Luhn et al. 2020). The amplitude of activity RVs ranges from tens of cm s^{-1} for the very quietest stars to hundreds of m s^{-1} for young and active stars. The timescale of this stellar variability ranges from minutes (acoustic modes; Chaplin et al. 2019) to hours (granulation), to days (rotation), to years (magnetic cycles; Fulton et al. 2015).

Over the last few years, a number of spectrometers have been commissioned that are stable at the level of several tens of cm s^{-1} . Some noteworthy examples include ESPRESSO (Pepe et al. 2021), MAROON-X (Seifahrt et al. 2018), EXPRES (Blackman et al. 2020), NEID (Schwab et al. 2016), and the Keck Planet Finder (KPF; Gibson et al. 2020). For such instruments, stellar activity is the dominant noise source for nearly all target stars. Several strategies have been developed for mitigating stellar activity. Most require dense sampling of RVs over timescales relevant to planetary and activity signals. Therefore, control of observational cadence has become even more critical as the community attempts to detect planetary signals below 1 m s^{-1} in the presence of stellar activity (Anglada-Escudé et al. 2016).

Achieving the desired observational cadence at any Doppler facility is a complex scheduling challenge. Most facilities support a number of active Doppler programs with different targets, observational priorities, and cadence requirements. In addition, most Doppler instruments share the telescope with other instruments with their own scheduling constraints (e.g., dark time for extra-galactic observations). Finally, weather losses are guaranteed but the set of impacted nights is not known in advance.

Today, the task of scheduling Doppler observations is performed almost entirely by humans. This is a challenging and time-consuming task. Schedulers must determine the sequence of observations that favor high program completion and equity among the supported programs. There is an opportunity for automated scheduling algorithms to save human effort and better achieve completion/equity goals. At present, we are aware of three Doppler facilities that are autonomously scheduled: the Las Cumbres Observatory (LCO) Network of Robotic Echelle Spectrographs (Siverd et al. 2018), the Calar Alto high-Resolution search for M dwarfs with exo-Earths with Near-infrared and optical Echelle Spectrographs (Quirrenbach et al. 2014), and the Automated Planet Finder Levy Spectrograph (Vogt et al. 2014).

Automated telescope scheduling has been deployed since at least the early 1990s. For a more detailed review of prior efforts, see Solar et al. (2016), Bellm et al. (2019a), Parazin et al. (2022), and references therein. A key early development was the Hubble Space Telescope’s SPIKE software (Johnston & Miller 1994), which has been adapted to suit other facilities including the James Webb Space Telescope (Giuliano & Johnston 2008). Many of the scheduling algorithms developed in the 2000s adopted “local search” or “greedy” algorithms. Here a schedule is built sequentially by selecting the next observation that contributes the most to a merit function. Such methods are not well suited to Doppler work because they lack the look ahead needed to schedule observations over long intervals.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

A significant recent development was the introduction of integer linear programming (ILP) methods to the telescope scheduling problem by Lampoudi et al. (2015) who developed the dynamic scheduling algorithm for the LCO network. ILP and mixed-integer linear programming (MILP) formulations of scheduling problems are common in the operations research communities and powerful commercial software libraries exist to solve these problems to their global optima. The ability to confidently find global optima makes these methods promising for the long-range planning needs of Doppler surveys. ILP- or MILP-based schedulers have since been developed for the Atacama Large Millimeter/Submillimeter Array (ALMA) by Solar et al. (2016) and the Zwicky Transient Facility by Bellm et al. (2019b).

This paper describes our efforts to automatically schedule Doppler observations at Keck Observatory. We provide some background on the prior human scheduling at Keck Observatory (Section 2). We describe our scheduling algorithm (Section 3). We present, apply and assess the performance of our scheduler during observations at Keck observatory during 2023 (Section 5). Finally, we conclude and offer some thoughts on future development (Section 6).

2. Manual Scheduling at Keck

Doppler searches for extrasolar planets have been conducted at the 10 m Keck I telescope with the High Resolution Echelle Spectrometer (HIRES) instrument since 1994 (Marcy & Butler 1996). The telescope is classically scheduled. principal investigators (PIs) submit proposals to a number of time allocation committees who award full or fractional nights over a six-month observing semester. Since 2010, the Doppler programs have received ≈ 40 –50 nights per semester distributed over ≈ 60 –70 full or fractional nights. Generally, awarded time has been anchored by a small number of large, multiselector projects, such as NASA mission support of Kepler, K2 and TESS, and a larger number of smaller 1–3 night projects. The set of all accepted programs (Doppler and other) are sent to the Keck Observatory with a cover sheet that includes basic scheduling constraints and preferences. Those requests include a set of nearly consecutive nights and at least one night per month to monitor short- and long-period planetary orbits. A human scheduler at Keck observatory heuristically balances these requests to determine a full semester schedule for the telescope.

Most PIs of Doppler programs at Keck elect to execute their observations with the infrastructure of the California Planet Search (CPS; Howard et al. 2010). CPS runs an ad hoc queue which schedules observations and maintains a pool of trained observers to execute the observations. CPS also reduces, curates, and distributes data. By collaborating with CPS, PIs may execute observations over a larger number of nights and share in weather losses. Indeed, many of smaller programs could not achieve scientifically useful cadence without access to a large number of nights through CPS.

Between 2010 and 2020, CPS observed $\approx 40,000$ stellar spectra (not counting multi-shot exposures) or an average of 2000 observations per semester. Each semester’s scheduling challenge is substantial as there are, in principle, $\mathcal{O}(N!) \sim 10^{6000}$ possible orderings. The CPS scheduler collects the observation requests at the start of the semester and before each night of observing. On the day preceding observations, the scheduler manually marks and removes previous observations.

The schedule is balanced each night with fractions of each program as a function of overall allocations. This helps balance weather losses, especially when entire nights are lost and each program loses time proportionally. The scheduler must also balance the distribution of target R.A. so that the telescope is not over- or under-subscribed at any point in the night. Follow up of exoplanet systems in the ecliptic further complicate the scheduling with the need to avoid pointing near the moon.

After the scheduler determines upcoming night’s targets, they generate a script, an ordered sequence of observations to be executed at the telescope. The scheduler also minimizes overheads associated with long slews and cable wrap limits. This task is analogous to the traveling salesman problem with the additional complications of target visibility windows and time-dependent slew times. The scheduler must have an intuitive understanding of the relationship between the R.A./ decl. coordinate system (that specify the targets) and the altitude/azimuth coordinate system (that specifies the telescope orientation). The relationship between these coordinate systems is a time-dependent and nontrivial transformation requiring tens of nights of observing experience to understand for planning purposes. The final schedule is slightly overfilled relative to what can be achieved even under exceptional observing conditions. At the telescope, CPS observers make real-time decisions to skip targets given current conditions and their knowledge of the scientific needs of the portfolio of programs.

The human effort required to schedule CPS observations is substantial. At the beginning of a semester, with the most targets to consider, the ratio of scheduling planning versus observation can be as high as one-to-one, and is reduced to one-to-ten by the end of the semester. PIs of contributing programs must also monitor the progress of their programs and request changes, if needed. By any standard, the amount of human time required to plan effectively while scheduling up to 100 stars in a night is high.

3. Scheduling Algorithm

Determining the precise timestamps of 1000 exposures throughout every night in an observing semester is an impractical feat when both problem size and weather uncertainty are considered. Precise minute-to-minute scheduling is only necessary for an upcoming night or small sequence of nights. However, long-range look ahead at the semester scale remains critical to the scientific goals of the algorithm. To balance these needs, our scheduling algorithm is hierarchical, splitting the problem into two stages.

First, it solves the scheduling problem at quarter night resolution, assembling groups of targets into each allocated quarter night while incorporating constraints on time, visibility, and cadence. The targets do not have a specified ordering within each quarter during this stage. In the second step, a small sequence of the nearest upcoming quarters is scheduled exactly with the targets assigned by the first step. In this paper, we describe the long-range semester scheduling stage in detail, including the initialization data, model construction, and objective function. The second step, which we refer to as the “Traveling Telescope Problem” (TTP) is described in detail in Handley et al. (2024).

Table 1
Sample of KPF Requests

Star Name	R.A. hr	decl. °	n_r^{inter}	n_r^{intra}	τ_r^{inter} Day	τ_r^{intra} hr	τ_r^{exp} s	Program Code
KOI-4032	19.0	42.7	1	1	NaN	NaN	120	IB
K00701	18.9	45.3	5	1	15	NaN	1200	LW
K00117	19.8	48.2	20	1	1	NaN	1200	LW
K00319	18.8	43.9	18	1	2	NaN	1400	JZ
T006324	22.1	67.5	14	3	1	1.5	900	FD

Note. From top to bottom: a noncadenced/single-shot observation, a low-cadence observation, a medium-cadenced observation, and a high-cadenced observation with three observations requested per night.

3.1. Inputs to the Scheduler

We begin the semester scheduling with a list of N_r many requests $\{1, \dots, N_r\}$, where r is the request index. Each request has the following attributes:

1. Star name [string].
2. Celestial coordinates [R.A., decl., epoch].
3. Maximum number of unique nights of observation over the semester [integer], n_r^{inter}
4. Maximum number of exposures per scheduled night [integer] n_r^{intra} .
5. Minimum duration between observations on different nights in days, the internight spacing [integer], τ_r^{inter} .
6. Minimum duration between observations on the same night in hours, the intranight spacing [float] τ_r^{intra} .
7. Exposure time τ_r^{exp} in seconds [integer]
8. Program code [string] that associates each request with an approved telescope program.

Table 1 shows a sample of KPF requests for the 2023B semester to illustrate the form of the input data.

The scheduler also requires a list of KPF quarter nights. If N_s is the total number of these allocated quarters, we index the list of slots arranged by increasing time $\{1, \dots, N_s\}$ with the letter s . Each individual awarded quarter night s need not be adjacent in the Keck schedule (in fact they are rarely so). We denote the duration of each slot in seconds with I_s .

Finally, since the ILP algorithm described below partitions observations into slots while satisfying the cadence constraints, we need to specify which targets can be observed on which slots. This is encoded in the ‘‘Visibility Matrix’’ V_{rs} .

1. $V_{rs} = 1$ if the target r is visible during quarter night slot s , and 0 otherwise.

During the 2023B semester, we calculated V_{rs} directly from the target coordinates, following a simple set of rules described in Section 5.1. However, V_{rs} may be specified according to an arbitrary set of rules, such as maximum air mass or minimum moon distance. We constrain the scheduling of targets only to when they are visible in Section 3.4.

Finally, the schedule requires an array that maps each slot s to a specific calendar date. This is done with t_s which is set to be 1 on the first day of the observing semester. The time difference (in days) between two slots s and s' is simply $\delta = t_{s'} - t_s$. We denote the maximum separation of any two given nights as $\delta_{\text{max}} = t_{N_s} - t_1$, the duration of the observing semester. We summarize the symbols from the main body of this text in [Appendix](#).

3.2. Weather Sampling

Variable weather at Maunakea impacts the nightly schedules. Some nights, cloud cover or poor seeing leads to significant reductions of throughput, rendering our target list practically unachievable. On severe weather nights, we are forced to forfeit observing altogether. While we cannot predict the precise extent of weather-related disruptions in advance, failing to incorporate amortized weather losses into our scheduler would introduce a substantial error into our completion expectations.

To address these challenges, we reserve 30% of all allocated quarter nights for adverse weather conditions. During these reserved periods, no observational targets are scheduled. This 30% allocation applies to a random selection of time slots in $\{1, 2, \dots, N_s\}$. In other words, for each slot s' randomly designated for weather-related downtime, we ensure that all observing requests r have $V_{rs'} = 0$.

We chose to sample entire quarters as opposed to sampling out 30% of the time within every quarter night because we believe this to be closer to the reality of weather losses. In general, individual quarters are usually either lost entirely to weather, or not. This works as a first order correction to our forecasting, and we discuss how this might be improved in Section 6.

3.3. Decision Variables

Our ILP formulation of the semester scheduling problem uses the following binary decision variables:

1. $Y_{rs} = 1$ if request r is scheduled to the quarter night slot s , and 0 otherwise. This variable specifies the full semester schedule, with the caveat that requests within a slot are not (yet) ordered. It has size $N_r \times N_s$.
2. $B_{rs's'} = 1$ if requested target r is scheduled to *both* slots s, s' , and 0 otherwise. This variable tracks the assignment of pairs of observations. It has size $N_r \times N_s^2$.
3. $D_{r\delta} = 1$ if requested target r is observed twice with time separation δ in days, and 0 otherwise, tracking the cadence achieved for each request. It has size $N_r \times (\delta_{\text{max}} + 1)$.

Figure 1 is a schematic of showing how Y_{rs} specifies a schedule. Figure 2 is a schematic showing the relationship between the three binary decision variables listed above.

3.4. Constraints

We achieve the desired observational cadence by enforcing the following constraints in our optimization.

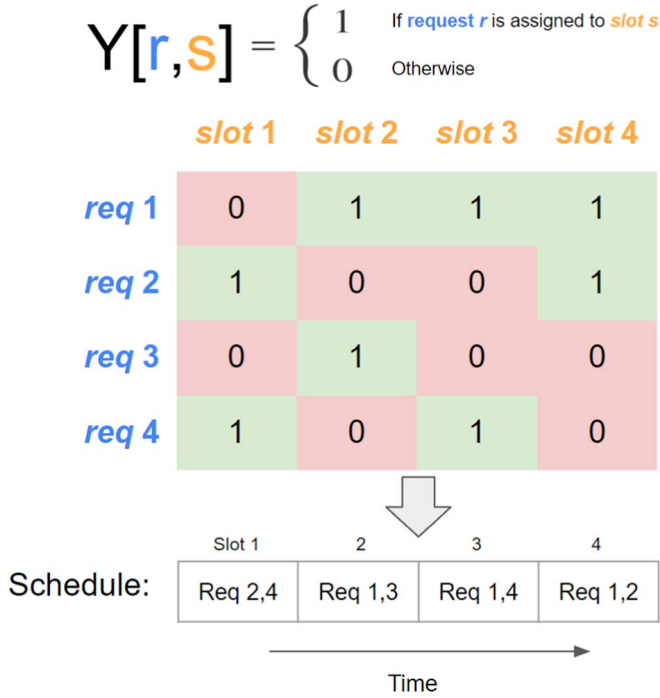


Figure 1. The Y_{rs} matrix specifies the set of requests assigned to each slot. Note that a single request may be scheduled multiple times and a slot may contain multiple requests.

Constraints 1, 2, 3: establish relationship between B and Y . For each request r , the values in $B_{r_{ss'}}$ act as *and* statements: values in B are only “turned on” (set to 1) when both corresponding values in Y also contain the value 1. To avoid degenerate constraints, we require $s < s'$. This is accomplished by:

$$B_{r_{ss'}} \leq Y_{rs} \quad (1)$$

$$B_{r_{ss'}} \leq Y_{rs'} \quad (2)$$

$$B_{r_{ss'}} \geq Y_{rs} + Y_{rs'} - 1 \quad (3)$$

$$\forall r = 1, \dots, N_r$$

$$\forall s = 1, \dots, s' - 1$$

$$\forall s' = 1, \dots, N_s.$$

Constraints 1 and 2 ensure that the values in $B_{r_{ss'}}$ cannot be 1 when either Y_{rs} or $Y_{rs'}$ is zero. Constraint 3 ensures $B_{r_{ss'}} = 1$ when both Y_{rs} and $Y_{rs'}$ are 1. Constraint 4: establish relationship between D and B . Values in D are 1 if a pair of scheduled observations exist for a request r at two slots separated by δ . That is, any pair of slot indices in B that satisfy this separation condition will force D to 1:

$$D_{r\delta} \geq B_{r_{ss'}} \quad \forall r = 1, \dots, N_r \quad \forall \delta = 0, 1, \dots, \delta_{\max} \\ \forall s = 1, \dots, s' - 1 \quad \forall s' = 1, \dots, N_s \\ \text{if } t_{s'} - t_s = \delta. \quad (4)$$

Constraint 5: targets may only be scheduled when they are visible. Values in Y must be forced to 0 whenever V holds the value 0.

$$Y_{rs} \leq V_{rs} \quad \forall r = 1, \dots, N_r \\ \forall s = 1, \dots, N_s. \quad (5)$$

If V_{rs} is 1, Y_{rs} may be either 1 or 0. The value is determined as part of the optimization. Constraint 6: Schedule may not exceed the maximum number of requested observations. The sum of the values in each row of the Y_{rs} matrix is the scheduled number of observations for request r , and should not exceed n_r^{inter} .

$$\sum_{s=1}^{N_s} Y_{rs} \leq n_r^{\text{inter}} \quad \forall r = 1, \dots, N_r. \quad (6)$$

Constraint 7: total exposure times must not exceed length of slot. We constrain the sum of each column of Y multiplied by the respective exposure time τ_r^{exp} and the number of exposures on each night n_r^{intra} of each index.

$$\sum_{r=1}^{N_r} Y_{rs} \tau_r^{\text{exp}} n_r^{\text{intra}} \leq I_s \quad \forall s = 1, \dots, N_s. \quad (7)$$

Note that this summation does not give any consideration to slew times or overheads, slightly overfilling some slots. We choose to oversaturate those in the upcoming night, in the next constraint.

Constraint 8: load the upcoming quarter night(s) with exposures. Should weather permit ideal observing conditions, slots within the upcoming observing night should be nearly filled with exposures. Let s^e and s^l be the earliest and latest allocated quarters in the upcoming night, respectively. Let f be the minimum fraction these quarters must be filled, i.e., the lower bound to the column sum.

$$\sum_{r=1}^{N_r} Y_{rs} \tau_r^{\text{exp}} n_r^{\text{intra}} \geq I_s f \quad \forall s = s^e, \dots, s^l. \quad (8)$$

We allow the TTP optimization to throw out high slew targets from upcoming nights, if necessary. We note that s^l can arbitrarily be chosen to saturate several nights at once, if one wanted to prepare multiple schedules in advance. Constraint 9: ensure minimum specified internight spacing of observations. Finally, we enforce the minimum cadence threshold for each request by restricting the values in D , in turn restricting certain pairs of observations.

$$D_{r\delta} = 0 \quad \forall \delta = 0, \dots, \tau_r^{\text{inter}} - 1 \\ \forall r = 1, \dots, N_r. \quad (9)$$

3.5. Objective Function

The objective of the autoscheduler is to maximize the number of observations, while penalizing long gaps between observations:

$$\text{Max} \left(\sum_{r=1}^{N_r} \sum_{s=1}^{N_s} Y_{rs} - C \sum_{r=1}^{N_r} \sum_{\delta=1}^{\delta_{\max}} \delta D_{r\delta} \right) \quad (10)$$

where C is a small enough constant that the second term is less than unity. We determined C using an upper bound on the value of the second term, assuming that all observations are achieved and any given pair of observations occur a distance δ_{\max} apart:

$$\frac{1}{C} \approx \sum_{1 \leq r \leq N_r, n_r^{\text{inter}} > 1} \binom{n_r^{\text{inter}}}{2} \delta_{\max} \approx 10^6. \quad (11)$$

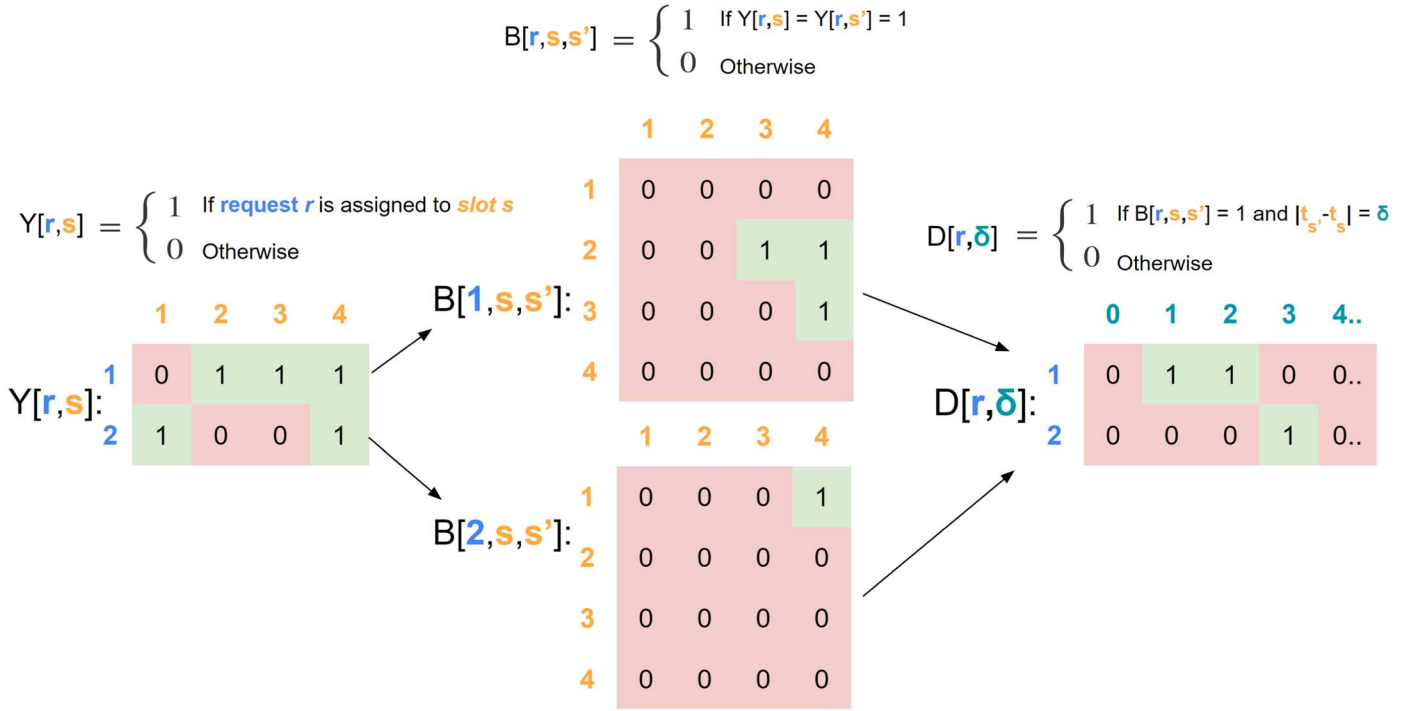


Figure 2. Graphical representation of binary decision variables. Left: Y_{rs} for the first two targets in Figure 1. To simplify the diagram, we show a scheme with one slot per night where slots occur on sequential nights. Our implementation at Keck uses up to four slots per night and slots are not guaranteed to occur on sequential nights. Middle: $B_{rs'}$ specifies any instance when a single request is scheduled into two separate slots, s and s' where $s < s'$. The upper matrix shows how $B_{1,s,s'}$ follows request 1 in $Y_{1,s}$; the lower matrix corresponds to request 2. Entries in B at and below the diagonal are always zero. Right: $D_{r,\delta}$ records whether two observations of request r were scheduled δ days apart.

This ensures that our cadence penalizing term never impedes on the observation of additional targets.

A simple interpretation of the objective is that the optimization algorithm will prioritize a family of solutions to the ILP that have the highest number of completed requests, then search these solutions locally for a schedule that achieves the densest possible sampling for every target. The unconstrained values of D will preferentially be set to 0, since they contribute negatively to the objective function.

We solve this ILP formulation using Gurobi version 10.0.1, a state-of-the-art optimization suite that solves ILP problems using the branch-and-bound algorithm (Gurobi Optimization, LLC 2023). We process observing requests and allocations using the Python programming language, and generate the ILP model within the Gurobi Python API. In our tests, Gurobi solves the model in ten minutes or less on a modern laptop (see Section 5.2 for further details).

4. Updating the Model Throughout the Semester

We rerun our algorithm after each observing night by integrating past observing data into our formulation as constraints. The result is a dynamic optimization scheme where a unique semester plan is generated under the new conditions each night.

4.1. Changes to the Formulation

Updating the environment of the semester problem requires the addition of new constraints and the modifications of others. We first construct a new matrix P_{rs} which contains the ingested observing data from all nights preceding the current night.

Recall that s^e is the first slot in the upcoming night. P will contain entries for all r for times before this value:

1. $P_{rs} = 1$ if request r was already observed during slot s , and 0 otherwise.

Constraint 10: at all time values before the present, P should dictate the behavior of Y .

$$Y_{rs} = P_{rs} \quad \forall r = 1, \dots, N_r \\ \forall s = 1, \dots, s^e. \quad (12)$$

In other words, the columns of Y that occurred in the past are included as “variables,” but are not freely optimized. They are used only to set the initial conditions on cadence for the remainder of the schedule. For this reason, it is important to consider that observers may choose to execute observations that violate constraints imposed on the model. Care must taken to remove the corresponding constraints, or else the values retrieved in P_{rs} may cause the model to become infeasible.

For example, on a night of notably poor weather conditions, observers might abandon the queue schedule to observe the brightest targets in the request list. Say that one of those chosen targets, r , was observed two days ago, but $\tau_r^{\text{inter}} = 5$. Observers choose to execute this observation, because it is one of the only targets bright enough for the current conditions. The next day, P_{rs} will contain two observations with $\delta = t_{s'} - t_s = 2$, which violates Equation (9) as a result of Equations (3), (4), and (12). This immediately makes the entire optimization infeasible.

We modify D to not track pairs of slots which have both passed. We redefine the bounds of Equation (4) as follows:

$$D_{r\delta} \geq B_{rss'} \quad \forall r = 1, \dots, N_r \quad \forall \delta = 0, 1, \dots, \delta_{\max} \\ \forall s = 1, \dots, s' - 1 \quad \forall s' = s^e, \dots, N_s \\ \text{if } t_{s'} - t_s = \delta. \quad (13)$$

For pairs of slots that both occur in the past, B cannot force D to be 1. The model will ignore violations to Equation (9) that occurred in the past. The minimum cadence will still be respected for all future values of Y .

Next, we restrict the domain of Equations (5) and (7) to $s = \{s^e, \dots, N_s\}$. The scheduler need only impose these rules on nights which have not already occurred, and must accept the values in P_{rs} even if they violate the rules enforced on upcoming nights. The adaptation to (5) will prevent model infeasibility if a request r was observed during slot s , but the value of V_{rs} was computed by the scheduler to be 0. The change to (7) catches the rare case where a past night achieved more exposures than were on the queue schedule, causing the sum of exposures to exceed I_s if the constraint is enforced.

4.2. Reoptimization

We optimize the semester problem again, with the same objective function as in Section 3.5. The values in P_{rs} will act as hard constraints, and the optimizer will explore possible plans of action for future slots which maximize the objective. While the semester optimum may be altered after even the first observing night, this dynamic method searches out the best path moving forward. For slots between s^e and s^l , we use the TTP (Handley et al. 2024) to precisely order all the assigned requests and accommodate intranight cadence constraints, then share the schedule with the observing team.

5. Implementation at Keck Observatory

This automated queue system is deployed at Keck Observatory by the CPS coalition (Howard et al. 2010) for the 2023B semester on time allocated for the KPF. CPS combines a multitude of Doppler search programs, resulting in 1388 observations scattered across 124 different queue quarter nights in 2023B. Within these collective pools is typically a combination of high-cadence, low-cadence, and single-shot observation requests from over a dozen programs.

5.1. Request Visibility

Using the start and end Julian Dates of each quarter night, we calculate the duration of each interval I_s in minutes, and determine the nominal visibility of each target in the request catalog from the given celestial coordinates. For Keck, we require that the target be above 30° elevation, above the Nasmyth platform, and more than 30° from the moon. This binary condition is evaluated at every minute within the interval. To reduce the risk of overconstraining the slew optimization described in Handley et al. (2024) by packing the night too tightly toward the beginning or end of the quarter night, we require that observing conditions be met for some constant $a \times I_s$ minutes for V_{rs} to be one (Figure 3), where a can be adjusted as necessary.

It should be noted that it is trivial to make the visibility requirements unique for each request, such as for targets that require lower air mass observations. We compile the subset of

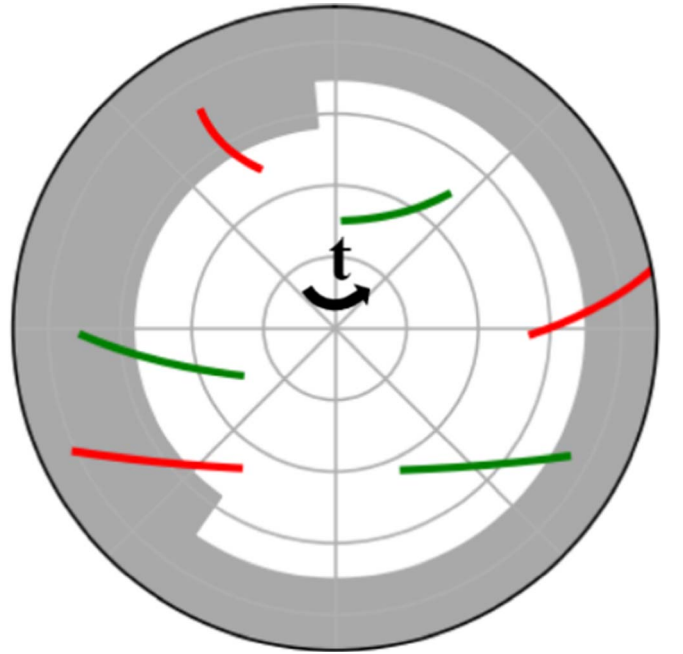


Figure 3. Sample paths in the altitude/azimuth frame at Keck. Targets that would be deemed visible have paths shaded in green as opposed to red. The fraction of time a for which requests must satisfy observing constraints is a free parameter. Here, we show the case where $a = 0.5$.

nights for which each request is visible at Keck, and construct the matrix V_{rs} .

5.2. Performance on KPF in 2023B

We present the first month of progress by our autoscheduler on the data set described in Section 3.1. This semester features 15 distinct programs with $N_r = 204$ unique requests for the pooled queue time of $N_s = 126$ scattered quarter nights. This totals to 1388 observations to schedule into 317 hr of allocated time. Early in the semester, the problem is largely unconstrained and runs on the order of 10 minutes to obtain a near optimal model. As observing nights pass, the computational load decreases. After one month, it routinely finds an optimal solution in under 3 minutes.

The scheduler produces various plots after each run. The first shows the completion rate of each program as a function of time. The current and forecasted progress of each program by the scheduler is displayed in Figure 4 subject to all constraints outlined in Section 3.4 and randomized weather sampling. In an ideal world, all programs would reach 100% at the last day of observations. However, due to mismatches between the distribution of allocated nights determined by the Keck Observatory scheduler (see Section 2), and the cadence needs of programs, achieving 100% completion is not always feasible.

The Keck Observatory scheduler determines the distribution based on limited information provided in the Keck coversheets, typically consisting of coordinates for the 10 highest priority targets. Consequently, the KPF allocations may not align well with the specific cadence requirements of all programs. A summary of the cadence distribution of these requests by program is shown in Figure 5.

For the entire semester, we also generate target-by-target schedules. Figure 6 provides an example for two targets from program LW, illustrating the visibility and scheduling of each

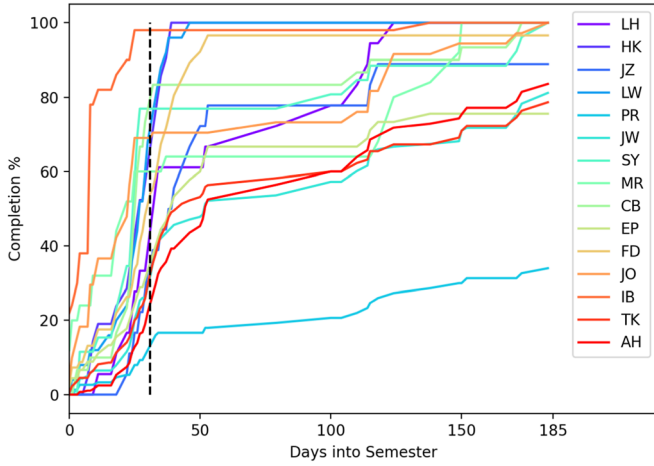


Figure 4. Cumulative progress of each program across the 2023B semester on KPF, as determined by our autoscheduler. The vertical dashed line indicates the most recent date the scheduler was run (one month of progress).

target. These plots show when each target is visible, how those periods coincide with allocated time on KPF, and when observations are consequently scheduled. We note that there is a significant difference between the nights available and the nights necessary to observe all targets to completion, which inhibits the progress of some programs. In 2023B, the KPF schedule was heavily front-loaded in the first two months (see Figure 6), which helped some programs achieve high forecasted completions, and hindered others.

Despite these challenges, eight of the fifteen programs in 2023B are expected to achieve 100% completion by the end of the semester. Four of the remaining programs are projected to achieve greater than 80% completion. Ultimately, only one program sits below 75% completion due to a mismatch in target coordinates and allocated KPF time.

The two targets in the PR program had limited visibility, rising primarily in the second half of 2023B. The target 26965 was requested for $n_r^{\text{inter}} = 100$ unique nights of observations. The other, 22049, was requested for $n_r^{\text{inter}} = 50$ unique nights of observations. As computed in V_{rs} , both targets were visible for a total of 37 unique nights, bounding the completion to a maximum of less than 50%.

The simulated schedule respects all requested minimum cadences, and attempts to schedule each consecutive observation at or near the minimum value. Long gaps between observations are sometimes required due to the distribution of allocated time, but are relatively uncommon. Across all targets, 158 pairs of consecutive observations occurred at exactly the requested cadence, and 57% of all pairs of observations occur within three days of the minimum cadence. The proportional deviations

$$\text{PropDev}(r, s, s') = \frac{t_{s'} - t_s - \tau_r^{\text{inter}}}{\tau_r^{\text{inter}}} \quad (14)$$

are summarized in the top right of Figure 5.

5.3. Performance on HIRES in 2023A

KPF is a newly commissioned instrument and nearly all early science programs in 2023B favor nightly cadence. We wish to demonstrate our algorithm’s suitability toward a wider

diversity of cadence needs. We further simulate our algorithm’s performance on target requests and the allocated time for CPS on Keck/HIRES in 2023A. HIRES in 2023A offers a more mature suite of programs consisting of a mix of high-, medium-, and low-cadence needs, as shown in Figure 5. This data offers a more robust opportunity to test our algorithms performance when cadence is the predominant scheduling constraint. Observations of this type are the most difficult for human schedulers to plan. Of the 543 observational gaps across all requests r , 111 occurred at the minimum allowed τ_r^{inter} , and 324 occur less than 5 days above τ_r^{inter} . The median violation of τ_r^{inter} is 3 days, with the proportional deviation shown in the bottom right of Figure 5. Additional target-by-target plots of this simulation are shown in Figure 7 for a subset of the program DG. Through this simulation, we find that our algorithm is capable of accommodating a mixed portfolio of programs with different cadence needs.

6. Future Directions

6.1. Support for Varied Weather Conditions

In principle, the autoscheduler operates at sufficient speeds (on the order of 5 minutes on a standard laptop) to support resolving the entire model as weather conditions at Keck evolve. However, these effects can be preemptively mitigated within the scheduling environment by producing multiple observing strategies based upon potential weather conditions for the upcoming night. Furthermore, these environments would benefit substantially from a more sophisticated weather sampling routine than that outlined in Section 3.2. We intend to equip future versions with a Monte Carlo sampler that weathers out nights based on historical W. M. Keck Observatory data.

6.2. Program Equity

An additional objective of queue systems is to promote equitable instrument access for programs of varying sizes and scientific goals. Some programs may have targets that have limited visibility, causing their scheduling to be disfavored. It may be desirable to equip the framework with variables that track the completion of each program, allowing the objective to homogenize these terms.

The autoscheduler presented here uses a linear objective function. An “equity” term may be added by adding the average absolute deviation of program completion may be added. Let N_{tot} be the total number of observations requested across all programs and N_p be the number requested by each program p , where the list of all programs is $\{1, \dots, p_{\text{max}}\}$. Let M_{rp} be a precomputed matrix that maps each request r to each program p .

1. $M_{rp} = 1$ if the target r is a subset of the program p , and 0 otherwise.

We use an additional variable A_p for this formulation, and define this variable with the following constraint:

1. A_p is the normalized (percentage) absolute deviation of each program from completion

$$A_p = \left| \frac{\sum_{r=1}^{N_r} \sum_{s=1}^{N_s} M_{rp} Y_{rs}}{N_p} - \frac{\sum_{r=1}^{N_r} \sum_{s=1}^{N_s} Y_{rs}}{N_{\text{tot}}} \right| \quad \forall p = 1, \dots, p_{\text{max}}. \quad (15)$$

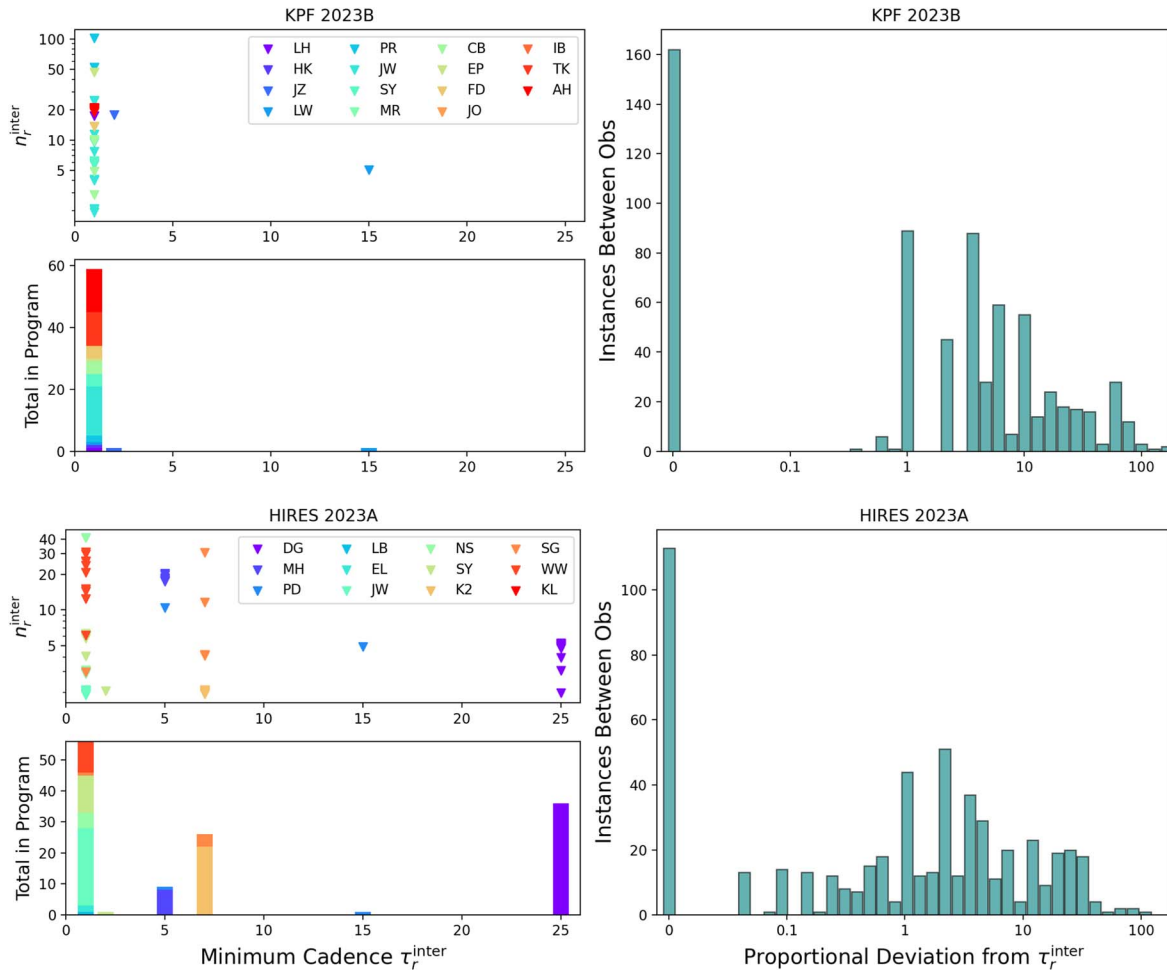


Figure 5. Top left: distribution of cadenced requests in the τ_r^{inter} and n_r^{inter} plane for KPF during the 2023B semester, with 5% induced scatter for visibility. The plot immediately below shows the cumulative number of requests in the respective programs at each cadence. Top right: histogram of how far the simulated schedule deviates proportionally from the ideal cadence for consecutive pairs of observations, across all requests and programs. The bottom three plots are the same as the top, but for HIRES in 2023A.

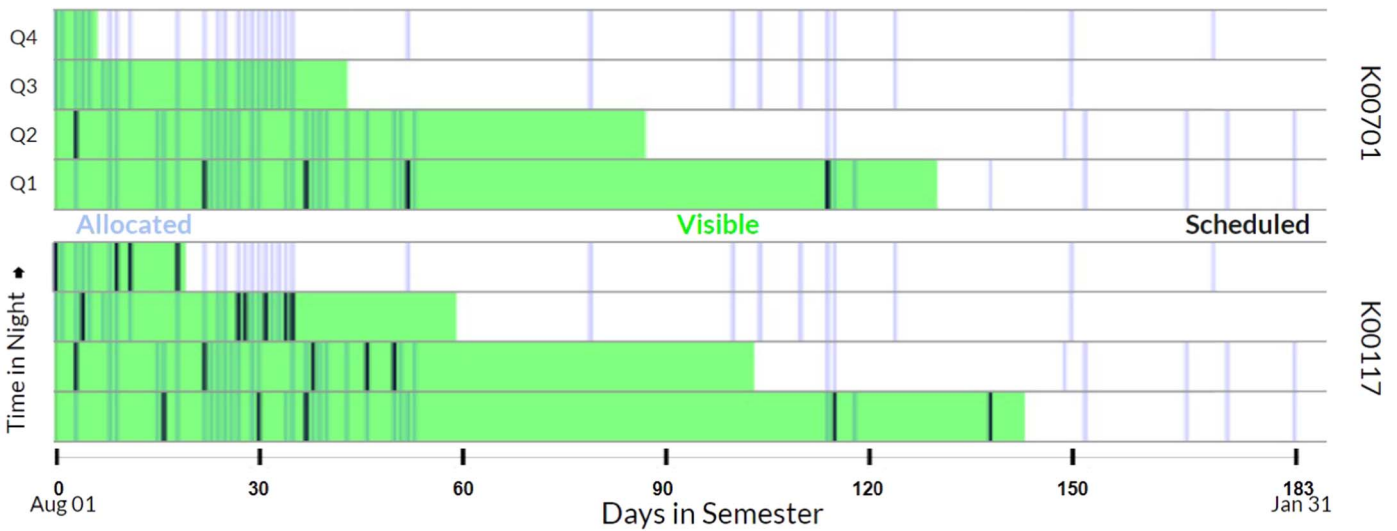


Figure 6. Visualization of the quarter night scheduling problem in 2023B for the two targets K00701 and K00117 (see Table 1), which are low- and high-cadence requests, respectively. Four rows correspond to each target, with time (quarter) increasing upward until resetting at the next plot. Top plot: visible quarter nights at Keck Observatory (green) throughout each night as a function of the date for K00701. Overplotted are the queue allocated quarter nights (blue) and the scheduled observations (black) for K00701 respecting $n_r^{\text{inter}} = 5$, $\tau_r^{\text{inter}} = 15$ as queried from the Y_{rs} variable. Bottom: same as top, but for the target K00117, respecting $n_r^{\text{inter}} = 20$, $\tau_r^{\text{inter}} = 1$.

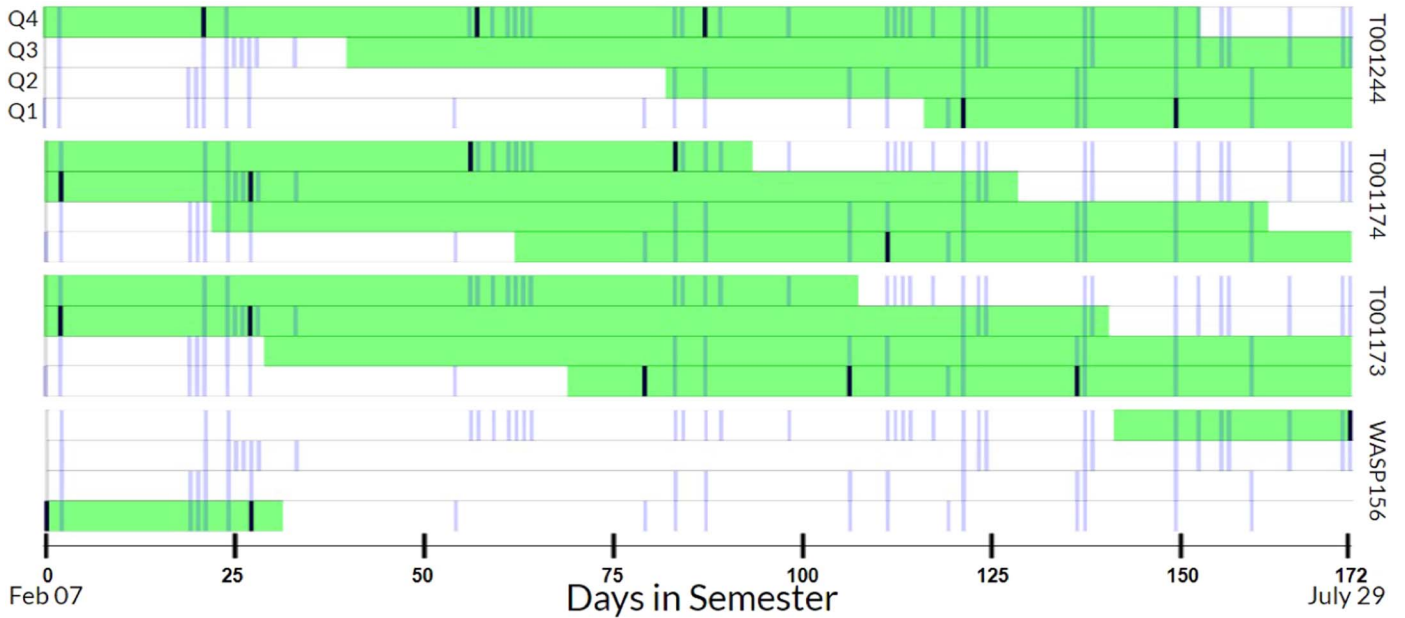


Figure 7. Select targets from our simulation of CPS Doppler observations on the HIRES instrument in 2023A, presented in the same format as Figure 6. Shown are four low-cadence requests with $n_r^{\text{inter}} = 5$, $\tau_r^{\text{inter}} = 25$, which are traditionally difficult to schedule. The last target has severely limited visibility, causing an outlier in the bottom right of Figure 5.

The objective in Section 3.5 can be modified to include the sum of these deviations for all programs times a scalar. However, solving with three objective terms reduces the interpretability of the ILP framework, since the scalar terms in the objective function need to be fine tuned by hand to find a balance of program equity and observational cadence. Setting a maximum acceptable deviation using the terms in A_p as a *constraint* on the model avoids this ambiguity.

7. Conclusion

In summary, we have developed an automated scheduler to manage the execution of approximately ~ 1000 Doppler observations during a single semester at the Keck Observatory. Our on-sky progress with KPF in 2023B demonstrates that the system consistently achieves high completion rates across various programs while adhering to internight and intranight cadence specifications. Additionally, our simulations on HIRES in 2023A showcase the system’s proficiency in handling diverse programs with varying cadence requirements, ensuring a significant portion of observations occur at the requested frequency.

Our scheduling system operates efficiently, capable of scheduling an entire semester within a few minutes. This not only saves considerable time for human schedulers but also provides a long-range outlook, alleviating CPS participating scientists from having to manually request targets for specific nights. The subsequent Traveling Telescope Problem optimization minimizes instrument downtime during observing runs, offering precise execution timing for each observation.

This algorithmic framework can be extended for use in any survey that involves collecting a large number of observations over numerous distinct observing intervals, particularly when

the cadence of observations is a critical factor. As the coming decades witness an increase in such endeavors, scheduling algorithms equipped to manage these challenges will become indispensable in achieving scientific objectives. Simulations employing this algorithm or variants can enhance our understanding of how allocated time distribution impacts different programs or surveys and how target selection can be optimized at ground-based observatories.

Acknowledgments

L.H., E.A.P., and J.L. acknowledge support from the Heising-Simons Foundation grant #2022-3832. V.V.M. acknowledges support from the UCLA Anderson School of Management. The authors thank the referee for their excellent comments which improved the clarity of this paper. Additionally, they wish to recognize and acknowledge the very significant cultural role and reverence that the summit of Maunakea has within the indigenous Hawaiian community. We are most fortunate to have the opportunity to conduct observations from this mountain.

Software: astropy (Astropy Collaboration et al. 2013, 2018), numpy (Harris et al. 2020), pandas (The pandas development Team 2020), gurobi (Gurobi Optimization, LLC 2023), matplotlib (Hunter 2007).





Appendix Variables

Table A1 lists the variables used in all preceding sections of this paper, along with their first usage.

Table A1
Symbols Used

Symbol	Definition	Section
a	Minimum portion of an interval for which a request must satisfy visibility constraints to be scheduled	5.1
A_p	Continuous variable indicating the absolute deviation of completion percentage of program p from the total semester completion percentage	6.2
$B_{rss'}$	Binary variable that indicates whether request r is assigned to both slots s, s' concurrently	3.3
C	Small constant used in the objective function to ensure the scheduling of additional observations is always prioritized	3.5
$D_{r\delta}$	Binary variable that indicates whether request r has multiple observations separated by an interval of δ days	3.3
f	Minimum fraction of time for which the upcoming nights should be filled with exposures	3.4
I_s	Duration of the slot s	5.1
M_{rp}	Precomputed program map. Holds 1 if request r is a subset of program p , 0 otherwise	6
n_r^{inter}	Maximum number of unique nights of observation over the semester for request r	3.1
n_r^{intra}	Maximum number of exposures per scheduled night for request r	3.1
N_p	Total observations requested by the program p	6.2
N_r	Total number of unique requests	3.1
N_s	Total number of allocated slots	3.1
N_{tot}	Total observations requested in the semester problem across all programs	6.2
P_{rs}	Past observation matrix. Holds 1 if request r has already been observed during the slot s , 0 otherwise	4.1
p	Program index	6.2
p^{max}	Number of unique programs	6.2
r	Request index	3.1
s, s'	Slot indices	3.1
s^e, s^ℓ	Earliest and latest slots indices which exist in the immediately upcoming night, respectively	3.4
t_s	Calendar date of the slot s relative to the first slot allocated (days)	3.1
V_{rs}	Visibility matrix. Holds 1 if request r is visible during the slot s , 0 otherwise	5.1
Y_{rs}	Binary variable that holds the scheduling state of request r to slot s . Holds 1 if request r is assigned to quarter night s , 0 otherwise	3.3
δ	Time difference (days)	3.1
δ_{max}	Calendar difference between the final and first slot in the semester (days)	3.1
τ_r^{exp}	Nominal exposure time of request r (seconds)	3.1
τ_r^{inter}	Minimum internight cadence between observations of the same request r (days)	3.1
τ_r^{intra}	Minimum intranight cadence between observations of the same request r (hours)	3.1

ORCID iDs

Luke B. Handley  <https://orcid.org/0000-0002-9305-5101>
 Erik A. Petigura  <https://orcid.org/0000-0003-0967-2893>
 Jack Lubin  <https://orcid.org/0000-0001-8342-7736>
 Howard Isaacson  <https://orcid.org/0000-0002-0531-1073>

References

- Anglada-Escudé, G., Amado, P. J., Barnes, J., et al. 2016, *Natur*, **536**, 437
 Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, *AJ*, **156**, 123
 Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, *A&A*, **558**, A33
 Bellm, E., Ford, E. B., Tohuvavohu, A., et al. 2019a, *BAAS*, **51**, 125
 Bellm, E. C., Kulkarni, S. R., Barlow, T., et al. 2019b, *PASP*, **131**, 068003
 Blackman, R. T., Fischer, D. A., Jurgenson, C. A., et al. 2020, *AJ*, **159**, 238
 Chaplin, W. J., Cegla, H. M., Watson, C. A., Davies, G. R., & Ball, W. H. 2019, *AJ*, **157**, 163
 Fulton, B. J., Rosenthal, L. J., Hirsch, L. A., et al. 2021, *ApJS*, **255**, 14
 Fulton, B. J., Weiss, L. M., Sinukoff, E., et al. 2015, *ApJ*, **805**, 175
 Gibson, S. R., Howard, A. W., Rider, K., et al. 2020, *Proc. SPIE*, **11447**, 1144742
 Giuliano, M., & Johnston, M. 2008, in Proc. Eighteenth Int. Conf. Automated Planning and Scheduling (Sydney: ICAPS), 107
 Gurobi Optimization, LLC 2023, Gurobi Optimizer Reference Manual, <https://www.gurobi.com>
 Handley, L. B., Petigura, E. A., & Misic, V. V. 2024, *ApJ*, **167**, 33
 Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Natur*, **585**, 357
 Howard, A. W., Johnson, J. A., Marcy, G. W., et al. 2010, *ApJ*, **721**, 1467
 Howard, A. W., Marcy, G. W., Bryson, S. T., et al. 2012, *ApJS*, **201**, 15
 Hunter, J. D. 2007, *CSE*, **9**, 90
 Johnston, M. D., & Miller, G. E. 1994
 Lampoudi, S., Saunders, E., & Eastman, J. 2015, arXiv:1503.07170
 Luhn, J. K., Wright, J. T., Howard, A. W., & Isaacson, H. 2020, *AJ*, **159**, 235
 Marcy, G. W., & Butler, R. P. 1996, *ApJL*, **464**, L147
 Parazin, B., Coughlin, M. W., Singer, L. P., Gupta, V., & Anand, S. 2022, *ApJ*, **935**, 87
 Pepe, F., Cristiani, S., Rebolo, R., et al. 2021, *A&A*, **645**, A96
 Quirrenbach, A., Amado, P. J., Caballero, J. A., et al. 2014, *Proc. SPIE*, **9147**, 91471F
 Schwab, C., Rakich, A., Gong, Q., et al. 2016, *Proc. SPIE*, **9908**, 99087H
 Seifahrt, A., Stürmer, J., Bean, J. L., & Schwab, C. 2018, *Proc. SPIE*, **10702**, 107026D
 Siverd, R. J., Brown, T. M., Barnes, S., et al. 2018, *Proc. SPIE*, **10702**, 107026C
 Solar, M., Michelon, P., Avarias, J., & Garces, M. 2016, *A&C*, **15**, 90
 The pandas development Team 2020, pandas-dev/pandas: Pandas, v1.4.1, Zenodo, doi:10.5281/zenodo.3509134
 Vogt, S. S., Radovan, M., Kibrick, R., et al. 2014, *PASP*, **126**, 359