

A Framework for Telescope Schedulers: With Applications to the Large Synoptic Survey Telescope

ELAHESADAT NAGHIB, PETER YOACHIM, ROBERT J. VANDERBEI, ANDREW J. CONNOLLY, LYNNE JONES

Abstract

Feature-Based telescope scheduler is an automated, proposal-free decision making algorithm that offers *controllability* of the behavior, *adjustability* of the mission, and quick *recoverability* from interruptions for large ground-based telescopes. It is easy to understand, implement and troubleshoot because of its coherent mathematical representation. Accordingly it can be modified by the astronomy community for context-dependent adjustments. This paper presents a generic version of the Feature-Based scheduler, with minimal manual tailoring, to demonstrate its potential and flexibility as a foundation for large ground-based telescope schedulers which can later be adjusted for other instruments. In addition, a modified version of the Feature-Based scheduler for the Large Synoptic Survey Telescope (LSST) is introduced and compared to previous LSST scheduler simulations.

Key terms and notations

fields	fixed partitions of the sky that can be covered in a single visit,
N_f	total number of the fields,
t	Greenwich Mean Time, in Modified Julian Date,
$\tau_s(t)$ ($\tau_e(t)$)	beginning (end) of the night that t lies in,
$\tau_{rise(set)}(i, f, t)$	rising (setting) time of field-filter (i, f) above (below) the acceptable airmass horizon at current night,
$\tau_n(i, f, t)$	time of the last visit of field-filter (i, f) before $\tau_s(t)$,
$id(t)$	ID number of the field that is visited at t ,
$ft(t)$	camera's filter at t ,
$n(i, f, t)$	total number of the visits of field-filter (i, f) before t ,
$slew(i, j)$	slew time from field i to field j in seconds,
$settling(i, j)$	mechanical settling time after slewing from field i to field j ,
Δt_f	time needed to change filter, a constant value about 2 minutes,
$t_{dome}(i)$	time needed to move the dome to make field i visible to the telescope,
$ha(i, t)$	hour angle of the center of field i at t in hours, $-12 \leq ha(i, t) \leq 12$,
$am(i, t)$	airmass of the center of field i at t ,
$br(i, t)$	brightness of the sky at the center of field i at t ,
$\sigma(i, t)$	seeing of the sky at the center of field i at t ,
$K(i, f, t)$	atmospheric extinction coefficient
W_1, W_2	given constant time window within which a revisit is valid,

1. INTRODUCTION

The Large Synoptic Survey Telescope (LSST), is the top national ground-based telescope according to the NASA 2010 Decadal Review. It is equipped with the largest digital camera that has ever been built for astronomy, and its unique scientific investigations range from the solar system to the edge of the optical universe. The revolutionary role of this telescope calls for no less than optimal operation.

During a night of operation, a ground-based telescope, is pointed to a direction of the sky, collects the incoming filtered light of the objects in that direction, and moves to another direction for the same task. The algorithm that makes the sequential decisions of which filter to use, and which direction to point the telescope to, is called *scheduler*. A scheduler has to maximize the scientific outcome of the telescope during its limited period of operation.

The mission of modern, large, ground-based telescopes, such as LSST, is constrained with various stochastic factors, and contains competing objectives which are vastly different in nature, due to the different nature of the scientific expectations. (See ([Ivezic et al. 2008](#)) for a detailed review of the LSST’s mission). In this paper, we propose a framework to formulate the problem of scheduling for the new generation of ground-based telescopes then introduce a scheduler based on the proposed model.

The first generation of schedulers for astronomical instruments were developed for space missions mainly to automate their operation due to their inaccessibility. ROSAT mission’s scheduler in 1990 ([Nowakowski et al. 1999](#)), Spike ([Johnston & Miller 1994](#)), Hubble Telescope’s scheduler in 1994, and HSTS ([Muscettola et al. 1995](#)) in 1995 are amongst pioneering developments on algorithmic scheduling of observations in the space missions.

Despite the similarity of the ultimate goal in the missions of space telescopes and ground-based telescopes, determining factors for the purpose of scheduling are fundamentally different. While space telescopes are required to respect kinematical and dynamical constraints, earth’s atmosphere is the main challenge in the scheduling of ground-based telescopes. The former is predictable and efficiently computable, while the later involves both inherent uncertainties, and uncertainties due to the computational limitations.

Earlier algorithmic approaches to the scheduling of ground-based telescopes are heavily based on observation *proposals* which are hand-crafted sequences of scripted astronomical observations. Proposals are generally tested only for feasibility, but not necessarily for optimality. For instance, the operation of Keck Telescope, 1993 ([Nelson et al. 1985](#)), is 100% based on proposals, and Hobby-Eberly Telescope, 1997 ([Shetrone et al. 2007](#)), has a semi-manual scheduling scheme.

Later, appearance of more expensive ground-based instruments with complex missions made it impossible to rely on hand-crafted proposals solely. Need for more efficient use of the instrument’s time led to the development of decision making algorithms for accessible facilities as well, and mainly to optimize their science output. For instance, scheduler of the Liverpool robotic telescope was designed in 1997 to automate and optimize the procedure of time allocation to chunks of scripted schedules. The time allocation strategy was preferred to the scheduling at the single visit level. The latter approach is referred to as optimum scheduling by the authors of ([Steele & Carter 1997](#)). The reason for choosing the time allocation scheme instead of optimum scheduling is stated to be the lack of recoverability of the latter choice in case of an interruption, because it potentially leads to excessive computational cost to reevaluate the sequence once it is interrupted. In this paper, we show that the scheduling in the single visit level can be quickly recovered in a memoryless framework, thus the optimality is not necessarily required to be sacrificed. Another example is the Las Cumbres Observatory Global Telescope Network (LCOGT) with one of the most advanced telescope scheduling algorithms ([Boroson et al. 2014](#); [Saunders & Lampoudi 2014](#)). LCOGT uses an integer linear programming (ILP) model, solved with the Gurobi algorithm ([Gurobi Optimization 2016](#)), to optimize the scheduling of observations over a global network of telescopes ([Lampoudi et al. 2015](#)). Due to the success of this approach the Zwicky Transient Facility

at Palomar Observatory (Bellm 2014) has also adopted an ILP scheduler¹. The ILP scheduling model performs well for observatories where slew overheads are small compared to exposure execution time. LCOGT is able to schedule observations in long contiguous blocks. In contrast, the LSST plans to have observations of about 30 seconds with slew times up to 2 minutes, hence it requires a scheduling algorithm that can explicitly minimize the slew times between successive observations. One could use the ILP scheduling approach to schedule large prescript blocks of observations for LSST. The blocks could be set to follow a path that only includes short slews (this is similar to the strategy taken in the scheduler developed by D. Rothchild et al.). The disadvantage of this approach and any other prescript schedule is principally the lack of recoverability from unpredictable conditions, for instance inability to dodge the clouds.

Still, in spite of the reliability of the fully automatic scheduling technologies, there are modern, expensive telescopes such as SALT, 2005 (Brink et al. 2008), and ALMA, 2013 (Wootten 2003), which are being operated based only on the traditional hand-crafted proposals. ALMA in particular, requires a highly regulated structure for proposals that potentially leads to suboptimality as discussed in (Alexander et al. 2017), where the authors suggest a number of corrections for the scheduling regulations to provide adaptivity to time-sensitive observations. However, even with occasional corrections, a verbal policy does not provide verifiable adjustability to the change of mission, as the operation of the telescope reveals newfound scientific desirables.

It seems that refusing to adopt the cutting-edge technology of fully automated scheduling for modern astronomical instruments is mainly due to the rooted notion of the observation proposals that perfectly served the scheduling of smaller, slower, and single-mission instruments in the past. This also could be a result of astronomer’s lack of trust toward the performance of the decision making machines in every levels of the scheduling. The LSST community, however, in addition to a proposal-based scheduler, introduced in (Delgado & Reuter 2016), have been supporting the design and implementation of proposal-free decision machines, such as Feature-Based scheduler, first introduced in (Naghib et al. 2016), as well as a variant of the fast marching army cadence by D. Rothchild et al. (2018, in preparation) that explore the possibility of a new generation of the schedulers for fast, multi-mission, big-data collecting instruments.

In Section 2, first we explain the choice of Markovian framework for the Feature-Based scheduler, and in Sections 2.1 and 2.2 we provide the mathematical details of the scheduler model in that framework. Section 3 presents two approaches for the optimization of the model’s parameters. Sections 4 demonstrates the application of the Feature-Based scheduler on the LSST which is then followed by a comparison between a modified version of the Feature-Based scheduler and LSST’s current official scheduler in Section 5. Finally, Section 6 presents our concluding remarks.

2. SCHEDULING FRAMEWORK

To run a multi-mission, ground-based telescope such as LSST, the scheduler has to offer *controllability*, *adjustability*, and *recoverability*.

- **Controllability:** The high-level mission objectives have to show sufficient variabilities with the variation of the model parameters. Otherwise, either the choice of the input information is irrelevant to the mission objectives, or the structure of the scheduler is degenerate. Controllability is necessary for a scheduler to be optimizable by the choice of the model parameters.
- **Adjustability:** For a complex, multi-objective mission it is common that the high-level objectives are required to be adjusted in the middle of the operational period. Adjustments take place according to updates of scientific desirables, or based on a feedback provided by the study of the system’s

¹ GitHub repository: https://github.com/ZwickyTransientFacility/ztf_sim

performance. Regardless of the reason for the adjustments, a scheduler must offer a systematic flexibility to be adjusted with a reasonable computational cost, and preferably no expert intervention. Hand-tuned scheduling strategies and verbal policies for instance, are not fully adjustable.

- **Recoverability:** The presence of unpredictable factors in the operation of ground-based telescopes are due to the natural stochastic processes (such as the weather), and complexity of the mechanical facility. Unscheduled downtimes and filter breakdowns are examples of the many unpredictable survey interruptions. On the other hand, there are inherently predictable interruptions, such as maintenance downtimes, and cable winding that also, due to the complexity of the mechanical system, are not computationally affordable and/or valuable to keep track of. Therefore, they are considered as stochastic variables as well. Given all the stochastic factors, a scheduler is required to be able to quickly make an alternative decision, once a previously unpredictable variable took a value. A scripted sequence of decisions for instance, lacks the recoverability attribute. Also, strategies with heavy look-back or look-ahead are not fast recoverables.

The Feature-Based scheduler is designed based on a Markovian model in which the flow of the input information, the decision procedure and it's relationship to the mission's objective are coherently expressed. Therefore, controllability of the scheduler is well-defined and verifiable. And it is adjustable, because of the Markovian structure that offers an explicit derivation of the design elements from high-level objectives, and finally is swiftly recoverable due to the inherent memorylessness of the Markovian Decision Process which for a decision at any time, only requires the current state of the system.

2.1. Markovian representation

Definition 1. Let $X(\cdot)$ be a stochastic process for which X_i represents the state of the system at t_i , and \mathcal{S} be the set of all possible states that the system can take. Let $\mathbb{P}(X_i)$, be the probability distribution of X_i on \mathcal{S} , then $X(\cdot)$ is a Markovian process, if and only if, it satisfies the following *memorylessness* property,

$$\forall i \quad \mathbb{P}(X_{i+1}|X_i) = \mathbb{P}(X_{i+1}|X_i, X_{i-1}, \dots, X_0),$$

where $\mathbb{P}(X_{i+1}|X_i)$ is the conditional probability distribution of the system's state at t_{i+1} given it's state at t_i , and $\mathbb{P}(X_{i+1}|X_i, X_{i-1}, \dots, X_0)$ is the conditional probability distribution of the system's state at t_{i+1} , given all of the states that the system has been in until t_i .

Memorylessness property asserts that the system's next state only depends on its current state and is independent of its earlier history. This property, is in fact, the main reason for choosing a Markovian framework for the scheduler.

Definition 2. Let $\langle \mathcal{S}, \mathcal{A}, P_a(\cdot, \cdot), R_a(\cdot, \cdot), \gamma \rangle$, be a *Markovian Decision Process (MDP)*, where \mathcal{A} is the set of actions, and $P_a(x, y)$ transition probability from state x to y which is equal to $\mathbb{P}(X_{i+1} = y|X_i = x, a)$, conditional probability of transition from state x to state y given action $a \in \mathcal{A}$. Similarly the transition reward is denoted by $R_a(x, y)$, and $\gamma \in (0, 1]$ is the discount factor.

Definition 3. Action $a_i \in \mathcal{A}$ is *admissible* for $\langle \mathcal{S}, \mathcal{A}, P_a(\cdot, \cdot), R_a(\cdot, \cdot), \gamma \rangle$, if it is *feasible*, thus it is possible to be taken at t_i , and *progressively measurable*, hence only dependent on the current state of the system X_i .

To control the system is to take an action at all decision steps t_i . For a Markovian control, the actions are required to be dependent only on the current state to preserve the memorylessness property of the closed-loop system, that's why we require the action to be progressively measurable. Notice that the decision steps $t_i \in \{t_j : 0 \leq j \leq N, t^N = T\}$, are not necessarily uniformly spaced, and are determined by the time that each transition takes. In this representation, t_0 is the start of the process and T is the finite time horizon of the process.

Definition 4. A deterministic policy $\pi : \mathcal{S} \rightarrow \mathcal{A}_i$, is a mapping from \mathcal{S} to the set of all admissible actions at t_i , denoted by \mathcal{A}_i .

A policy provides a time invariant control law that for all possible $x_i \in \mathcal{S}$ suggests an admissible action for transition to the next state, which automates the control of the system.

Definition 5. A deterministic optimal policy π is a solution to the following optimization problem,

$$\underset{\pi}{\text{maximize}} \quad E_{\pi} \left[\sum_{i=0}^N \gamma^i R_{\pi(X_{i-1})}(X_{i-1}, X_i) | x_0 \right], \quad (1)$$

where x_0 is a given initial state.

In other words, a deterministic optimal policy maximizes the expected discounted sum of the rewards. By discounting the later rewards versus earlier rewards through $0 < \gamma \leq 1$, we tune the priority of the overall gain versus instant gains.

Proposition 1. For the Markov decision process of $\langle \mathcal{S}, \mathcal{A}, P_a(., .), R_a(., .), \gamma \rangle$, there exists a deterministic optimal policy, and it can be written as follows,

$$\pi^* = \arg \min_{a_i \in \mathcal{A}_i} E[\Phi(X_{i+1}) | a_i], \quad (2)$$

where $\Phi : \mathcal{S} \rightarrow \mathbb{R}$ is a function.

Proof. Consider a function $C_{\pi} : \mathcal{S} \rightarrow \mathbb{R}$, defined as follows,

$$\begin{aligned} C_{\pi}(x_i) &= -E_{\pi} \left[\sum_{i \leq j}^N \gamma^{i-j} R_{\pi(X_j)}(X_j, X_{j+1}) | x_i \right] \\ &= -E_{\pi} [R_{\pi(x_i)}(x_i, X_{i+1}) | x_i] - \gamma E_{\pi} \left[\sum_{i+1 \leq j}^N \gamma^{j-(i+1)} R_{\pi(X_j)}(X_j, X_{j+1}) | x_i \right]. \end{aligned}$$

Then by applying the law of total expectation on the second term,

$$\begin{aligned} C_{\pi}(x_i) &= -E_{\pi} [R_{\pi(x_i)}(x_i, X_{i+1}) | x_i] - \gamma E_{\pi} \left[E_{\pi} \left[\sum_{i+1 \leq j}^N R_{\pi(X_j)}(X_j, X_{j+1}) | X_{i+1} \right] | x_i \right] \\ &= -E_{\pi} [R_{\pi(x_i)}(x_i, X_{i+1}) | x_i] - \gamma E_{\pi} [C_{\pi}(X_{i+1}) | x_i], \end{aligned}$$

and by assuming a finite state space,

$$C_{\pi}(x_i) = -E_{\pi} [R_{\pi(x_i)}(x_i, X_{i+1}) | x_i] - \gamma \sum_{x_{i+1} \in \mathcal{S}} \mathbb{P}(x_{i+1} | \pi(x_i), x_i, x_{i-1}, \dots, x_0) C_{\pi}(x_{i+1}).$$

Then by the Markov property,

$$C_{\pi}(x_i) = -E_{\pi} [R_{\pi(x_i)}(x_i, X_{i+1}) | x_i] - \gamma \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) C_{\pi}(x_{i+1}),$$

where, $P_{\pi(x_i)}(x_i, x_{i+1})$ is the transition probability from x_i to x_{i+1} under the action of $\pi(x_i)$. Now, let $C^*(x_i) = \min_{\pi} C_{\pi}(x_i)$ then,

$$\begin{aligned} C^*(x_i) &= \min_{\pi} (-E_{\pi}[R_{\pi(x_i)}(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) C_{\pi}(x_{i+1})) \\ &= \min_{\{\pi(x_i), \pi(x_{i+1}), \dots\}} (-E_{\pi}[R_{\pi(x_i)}(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) C_{\pi}(x_{i+1})) \\ &= \min_{\pi(x_i)} (-E_{\pi}[R_{\pi(x_i)}(x_i, X_{i+1})|x_i] - \min_{\{\pi(x_{i+1}), \pi(x_{i+2}), \dots\}} \gamma \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_{i+1}, x_i) C^{\pi}(x_{i+1})). \end{aligned} \quad (3)$$

Note that both of the one step reward $R_{\pi(x_i)}(x_i, X_{i+1})$ and transition probability $P_{\pi(x_i)}(x_i, X_{i+1})$ depend only on $\pi(x_i)$ which is the action taken at t_i . For the next time step, one can construct a \hat{C} function such that, $\hat{C}(x_{i+1}) = \min_{\pi(x_{i+1})} \min_{\pi} C^{\pi}(x_{i+1})$, then,

$$\begin{aligned} \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) \min_{\{\pi(x_{i+1}), \pi(x_{i+2}), \dots\}} C^{\pi}(x_{i+1}) &= \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) \hat{C}(x_{i+1}) \\ &\geq \min_{\pi} \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_{i+1}, x_i) C^{\pi}(x_{i+1}). \end{aligned}$$

On the other hand,

$$\sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) \min_{\{\pi(x_{i+1}), \pi(x_{i+2}), \dots\}} C^{\pi}(x_{i+1}) \leq \min_{\pi} \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) C^{\pi}(x_{i+1}).$$

Therefore,

$$\min_{\pi} \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) C^{\pi}(x_{i+1}) = \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) \min_{\{\pi(x_{i+1}), \pi(x_{i+2}), \dots\}} C^{\pi}(x_{i+1}).$$

By substituting the second term of the right hand side of Equation (3) with the right hand side of the above equation,

$$\begin{aligned} C^*(x_i) &= \min_{\pi(x_i)} (-E_{\pi}[R_{\pi(x_i)}(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) \min_{\pi} C^{\pi}(x_{i+1})) \\ &= \min_{\pi(x_i)} (-E_{\pi}[R_{\pi(x_i)}(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in \mathcal{S}} P_{\pi(x_i)}(x_i, x_{i+1}) C^*(x_{i+1})). \end{aligned}$$

The last equality follows from the definition of C^* , and is in the form of Optimal Bellman Equation, for which a solution exists [Bellman 1957], and $C^*(x_0) = \min_{\pi} C^{\pi}(x_0)$ attains the optimal value, by the construction of $C^{\pi}(x_0)$ which is equal to $-E[\sum_0^N R_{\pi(x_i)}(x_i, x_{i+1})|x_0]$.

Given a C^* , an optimal policy can be simply evaluated,

$$\begin{aligned} \pi^*(x_i) &= \arg \min_{a_i \in \mathcal{A}_i} (-E_{\pi}[R_{a_i}(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in \mathcal{S}} P_{a_i}(x_i, x_{i+1}) C^*(x_{i+1})) \\ &= \arg \min_{a_i \in \mathcal{A}_i} (-E_{\pi}[R_{a_i}(x_i, X_{i+1}) - \gamma C^*(X_{i+1})]|x_i]). \end{aligned} \quad (4)$$

Finally, define,

$$\Phi(X_{i+1}) := R_{a_i}(x_i, X_{i+1}) - \gamma C^*(X_{i+1}), \quad (5)$$

and substitute in Equation (4) in order to obtain Equation (2). \square

For the telescope scheduler we require the policy to be deterministic, because the simulations have to be repeatable for comparison and evaluation purposes, however it can be shown that the deterministic optimal policy is not only optimal amongst deterministic policies, but also is optimal amongst stochastic policies. Therefore the choice of deterministic policy does not harm the optimality of the control.

As a result of Proposition 1, the solution space of the optimization problem (1), can be reduced from search over the set of policies to search over Φ functions.

2.2. Markovian approximation

For a decision that is inherently time dependent, such as scheduling an observation, only a maximal definition of the system's state yields a perfect Markovian system, in which case, the state-space includes all of the possible decision sequences. In particular, LSST requires a sequence of about 1000 decisions at each night. Therefore, storing all possible scenarios requires a state space of size N_f^{1000} , that clearly is neither tractable nor storable in a realistic memory. In order to overcome the curse of dimensionality, we have designed a set of *features* to summarize the most important information required for the scheduling. Thus, the system of telescope-environment is only an approximated Markovian system, once its state-space is replaced by a *feature-space*.

On the other hand, despite the reduction of the search space from the mapping space to the function space, made possible by Proposition 1, Problem (1) is still an infinite dimensional optimization problem because its variable is a function. To be able to numerically compute the Φ function, we propose a parametrized function approximation for $\Phi(x_i)$,

$$\tilde{\Phi}_\theta(x_i) := \sum_{j=1}^m \theta_j \Phi_j(x_i),$$

where θ is the vector of variables that characterize $\tilde{\Phi}(.)$, and $\Phi_i(x_i)$'s are the *basis functions* that are designed to modify the features in order to incorporate the astronomical observational knowledge into the decision maker's structure. With this approximation, the space of search is reduced from the space of functions to a finite dimensional vector space. This approximation substitutes the original optimal policy (2) with the following approximate policy,

$$\tilde{\pi}_\theta^*(x_i) = \arg \min_{a_i} E[\tilde{\Phi}_{\theta^*}(X_{i+1})|a_i] = \arg \min_{a_i} \sum_{j=1}^k \theta_j^* E[\Phi_j(X_{i+1})|a_i], \quad (6)$$

where θ^* is a solution to the following optimization problem, in which policy π is fully determined by θ .

$$\underset{\theta}{\text{maximize}} \quad E_{\pi_\theta} \left[\sum_{i=0}^N \gamma^i R_{\pi_\theta(X_{i-1})}(X_{i-1}, X_i) | x_0 \right], \quad (7)$$

3. SCHEDULER OPTIMIZATION

Given the approximated optimal policy in Equation (6), the only remaining fundamental step to have a scheduler, is to find θ^* by solving Problem (7). The following two sections describe two different approaches to find a θ^* . The first approach requires a specific class of the high-level mission objective function, and is faster. The second approach is applicable for all types of the high-level mission objective functions, however requires more computational resources.

3.1. Reinforcement Learning

Assume that there exists a well-defined notion of the instant reward for each state transition, then by the definition of $\Phi(x_i)$ given in Equation (5) one has,

$$\begin{aligned}\Phi(x_i) &= -R_{\pi^*(x_{i-1})}(x_{i-1}, x_i) - \gamma C^*(x_i) \\ &= -R_{\pi^*(x_{i-1})}(x_{i-1}, x_i) - \gamma E_{\pi^*}[R_{\pi^*(x_i)}(x_i, X_{i+1}) + \gamma C^*(X_{i+1})|x_i] \\ &= -R_{\pi^*(x_{i-1})}(x_{i-1}, x_i) + \gamma E_{\pi^*}[\Phi(X_{i+1})|x_i].\end{aligned}\quad (8)$$

Equation (8) is valid when the system is operated by the optimal policy, π^* . Accordingly, for the parametrized Φ function, we require that,

$$\tilde{\Phi}_{\theta^*}(x_i) = -R_{\pi_{\theta^*}(x_{i-1})}(x_{i-1}, x_i) + \gamma E_{\pi_{\theta^*}}[\tilde{\Phi}_{\theta^*}(X_{i+1})|x_i]. \quad (9)$$

The main idea to find the best approximate $\tilde{\Phi}(.)$ is to start the process of the decision making with an arbitrary set of variables, θ^0 , and make the decisions according to Policy (6) with the associated $\tilde{\Phi}_\theta(.)$, then update the variables in each decision step, so that Φ 's linear approximation gradually respects Equation (9) for all $i \in \{j : t_0 \leq t^j \leq T\}$.

Note that at t_i , after the transition from x_{i-1} to x_i we have the value of $R_{\pi_{\theta^i}}(x_{i-1}, x_i)$ already evaluated in the decision making procedure, and $\Phi_{\theta^i}(x_i)$ can be approximated by $E_{\pi_{\theta^i}}[\Phi_{\theta^i}(x_i)]$ which is also evaluated in the decision making process, where θ^i is the last version of the optimization variables at t_i . Using the desired value given in Equation (9), the update is as follows,

$$\tilde{\Phi}_{\theta^{i+1}}(x_i) = (1 - \alpha)\tilde{\Phi}_{\theta^i}(x_i) + \alpha(\gamma E_{\pi_{\theta^i}}[\tilde{\Phi}_{\theta^i}(x_{i+1})|\pi_{\theta^i}(x_i)] - R_{\pi_{\theta^i}}(x_{i-1}, x_i)), \quad (10)$$

in which, $0 < \alpha < 1$ is the learning rate. The first term in the right hand side of the equation is the latest approximated value of the $\tilde{\Phi}$ function associated with θ^i with $(1 - \alpha)$ amount of contribution, and the second term is the value of $\tilde{\Phi}$ function according to Equation (9) with α amount of contribution. Clearly for smaller α 's this update imposes smaller adjustments. Accordingly, updates of the variables θ_j , $j = 1, \dots, k$ can be expressed as follows,

$$\begin{aligned}\theta_j^{i+1} &= \theta_j^i + (\tilde{\Phi}_{\theta^{i+1}}(x_i) - \tilde{\Phi}_{\theta^i}(x_i))\Phi_j(x_i) \\ &= \theta_j^i + \alpha(-\Phi_{\theta^i}(x_i) + \gamma E_{\pi_{\theta^i}}[\tilde{\Phi}_{\theta^i}(X_{i+1})|x_i] - R_{\pi}(x_{i-1}, x_i))\Phi_j(x_i).\end{aligned}\quad (11)$$

This Learning method is called *Temporal-Difference* (TD) learning with function approximation ([Tsitsiklis & Van Roy 1997](#)). Variants of this reinforcement learning method have been successfully applied to real-life problems such as training of a backgammon player ([Tesauro 1995](#)).

Note that to be able to use the TD reinforcement learning method, it is necessary to have a well-behaved notion of the reward that reflects the instant gain of any decision at all of the decision steps. Moreover, the discounted sum of the instant rewards has to reflect the objective of the mission. For instance, in the LSST scheduling problem, after each visit, the negative of the slew time, is a well-defined instant reward that reflects how time-efficiently the telescope is being used, however it does not reflect all aspects of the mission's objective such as, efficiency of the same night revisits (explained in Section 4), for which there is no equivalent instant reward. For this reason, we also implemented a global optimizer for the LSST scheduler that does not require a notion of the instant reward and directly optimizes the mission's objective function over a limited episode of the simulated scheduling.

3.2. Global Optimization

In the absence of a well-defined instant reward, instead of solving problem (7), The following problem can be solved:

$$\underset{\theta}{\text{maximize}} \quad U_{\pi_{\theta}}(x_i, x_{i+1}, \dots, x_j), \quad (12)$$

where $U_{\pi_{\theta}}(x_i, x_{i+1}, \dots, x_j)$ is a utility function that measures the performance of the scheduler on a simulated episode of the operation from t_i to t_j by policy π_{θ} . In this approach, a general $U(\cdot)$, can not be explicitly expressed as a function of θ , therefore a global optimizer that can maximize a black-box function is required. Evolutionary optimizers have successfully been applied to numerous real-life problems involving black-box function optimization, and specifically astronomical mission planning such as the scheduling of Exoplanet Characterisation Observatory (Garcia-Piquer et al. 2015). We used the *eDE* evolutionary optimizer (Naghib & Nobakhti 2016) which is an adaptive version of the Differential Evolution (DE) algorithm ((Storn & Price 1997)). DE is generally one of the most efficient evolutionary algorithms and the *eDE* variant uses a notion of entropy to automatically preserve the diversity of the candidate solutions. As a result, in contrast with DE, it does not require the user to choose any tuning parameters for the algorithm, which is the most time-consuming task in using an evolutionary optimizer. In addition, *eDE*, similar to any other evolutionary algorithm is highly parallelizable, and the computational time can be almost linearly decreased with respect to the number of computational cores.

4. PROBLEM OF SCHEDULING FOR LSST

The LSST's mission is to uniformly scan the visible sky within 5 different regions shown in Figure 1. Each region, also referred to as *survey*, has certain science-driven desirables and constraints, defined and precisely described in (Ivezic et al. 2008).

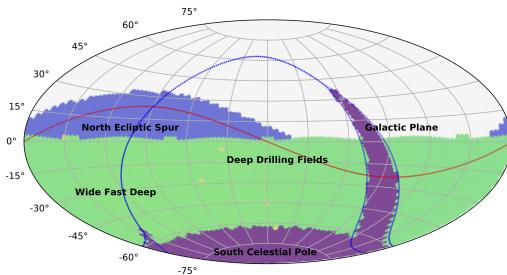


Figure 1. Regions of the sky with different requirements and constraints for scheduling: (1) Galactic Plane Region (GP), (2) Universal or Wide Fast Deep (WFD), (3) South Celestial Pole (SCP), (4) North Ecliptic Spur (NES), and (5) Deep Drilling Fields (DD).

The notion of the features, enables the scheduler to systematically fetch all of the various requirements and turn them into comparable quantities for the purpose of the decision making in the Markovian framework. The proposed feature-space of the LSST contains seven features, each can be evaluated given a field i , a filter f , and a time t . The fields discretize the visible sky through a fixed partitioning. Each field can be captured through a single visit, and there are 6 possible filters, $[u, g, r, i, z, y]$, for each visit. And finally the time domain is discretized by the natural timing of the process. Since a consecutive visit of the same field-filter is not allowed, there is a slew time between any two decisions, therefore $t_j - t_{j-1} > 0$. On the other hand, the operation is over a limited time horizon, T , thus the number of the decision time steps is finite. In conclusion, a finitely discretized sky, a finite number of the filters and a finite number of the time steps, pose a finite feature-space, denoted by $\{(f_1(i, f, t_j) \dots f_7(i, f, t_j)) : i =$

$1 \dots n_f, f \in \{u, g, r, i, z, y\}, j = 0, \dots N\}$. Then the implication of the policy, stated in Equation (6), for the LSST scheduler would be as follows,

$$\tilde{\pi}^*(x_j) = \arg \min_{(i,f) \in \mathcal{A}_j} \sum_{k=1}^5 \theta_k^* E_\pi[\Phi_k(X_{j+1})|x_j], \quad (13)$$

where $x_j = [f_{1,\dots,7}(id(t_j), ft(t_j), t_j)]$ is the 7-dimensional state at t_j , and (i, f) is a feasible pair of field-filter. Section 4.3 introduces the constraints under which a field-filter pair, (i, f) is feasible at t_j . Accordingly, \mathcal{A}_j is the set of all field-filter pairs that are feasible at t_j .

For a modular approach to the implementation of the scheduler, expected value of the basis functions, $E_\pi[\Phi_k(x_{j+1})|x_j]$ for $k = 1, \dots, 5$, are evaluated in separate modules. Those basis functions that address the environmental parameters are developed by the LSST community. For instance, see (Sebag et al. 2008) and (Sebag et al. 2007), for cloud cover measurements at the LSST site that were used to develop a cloud model and, see (Yoachim et al. 2016) for the sky brightness predictive model. For the details of the basis functions of the Feature-Based scheduler for LSST refer to Section 4.2.

The last step is to implement the training procedures (described in Section 3) on the LSST model to derive θ^* . Training of the LSST scheduler is explained in Section 4.4 with adopting two sample objective functions. However, the LSST community, and basically any individual can design their own mission objective function, whether it allows for well-defined instant rewards or not, then train the scheduler through the open source training code, and find a new set of θ^* that principally leads to a different behavior of the scheduler due to different objectives.

4.1. Features of the telescope scheduler

For designing the features, it is important to avoid redundancy and ensure the inclusion of determining information. It is also critical to hold a modular approach in the delivery of the information to the decision procedure. For instance, consider the amount of time, Δt , it takes for a telescope to move on from a visit. In the LSST problem, Δt mainly depends on the slew time, mechanical settling time, the dome placement time, and the time it takes to change the filter. All of these timings are available through a precise simulation of the LSST model (Delgado et al. 2014). A modular design would be to bring the summation of the operational timings to the stage of the decision, instead of bringing them separately as different features. This approach makes the implementation significantly simpler, and more readable, and in a conceptual level, makes it possible to track the effect of the operational timing in the overall outcome of the decision maker. Particularly, since the operational cost is independent of the amount to which each cause contributes to the overall Δt , bringing the timing of the separate procedures separately in the decision making level adds unnecessary complications to the design, and consequently, makes it hard to back track the output-input behavior of the scheduler.

This section proposes seven features for the description of the LSST-environment state in Table (1). They are designed to efficiently carry the determining information with a modular approach. Each feature is denoted by $f_k(i, f, t)$ for $k = 1 \dots 7$, and indexed by the triplet of (i, f, t) , field, filter, and time. To make a decision at t , the scheduler computes principally all of the seven features for all of the (i, f) pairs, however, there are some features that don't change in every time step, for instance, if i is not visited at t^j , then $f_5(i, f, t^j) = f_5(i, f, t^{j+1})$. For such cases, the implementation has a categorized updating procedure to avoid redundant computations.

4.2. Basis functions of the telescope scheduler

Basis functions are fully determined by the value of the features, and are denoted by $\Phi_k(f_1(i, f, t), \dots, f_7(i, f, t))$, for $k = 1 \dots 5$. Each Φ_k , is indexed by a triplet of (i, f, t) , time, field, filter, and is evaluated at every decision making step, t_j , for all field-filter pairs (i, f) . Similar to the update procedure of the features,

Table 1. Features of the approximated Markovian model for the telescope-environment system. Features $f_1 \dots f_7$ provide a memoryless, approximate description of the system's state.

Notation	Definition\Description
$f_1(i, f, t)$	$(slew(id(t), i) + settling(id(t), i) + \Delta t_f I_{ft(t) \neq f} \vee t_{dome}(i))$: either the time required to point the telescope to i , and change the filter to f , or the time required to relocate the dome to make i visible. Whichever that is larger.
$f_2(i, f, t)$	the total number of the same-night visits of field-filter (i, f) until t .
$f_3(i, f, t)$	$(t - \tau_n(i, f, t))I_{\{\theta(i, f, t) > \tau_s(t)\}}$, time since the last same-night visit of (f, i) .
$f_4(i, f, t)$	remaining time for field-filter (i, f) to become invisible, either by passing the airmass or the moon-separation threshold, or being covered by temporary objects such as clouds, as projected at t .
$f_5(i, f, t)$	co-added depth, a measure of cumulative quality of past visits of field-filter (i, f) until t .
$f_6(i, f, t)$	5σ -depth, a measure for quality of visiting field-filter (i, f) at t , depending on seeing, sky brightness, and airmass. $f_6(i, f, t) = C_m + 2.5 \log(\frac{0.7}{\sigma(i, t)}) + 0.50(br(i, t) - 21) - K(i, f)am(i, t)$ where C_m is a scaling coefficient.
$f_7(i, t)$	hour angle of field i at t .

for a decision at time t_j , all six basis functions should be evaluated for all pairs of (i, f) , except for the field-filters that are infeasible at t_j . Feasibility of a field-filter can be evaluated after the evaluation of the features, by applying the constraints of the region that the field belongs to (see Section 4.3 for the list of constraints). Thus, while it is required to evaluate the features for all possible pairs of (i, f) at all decision steps, the number of the basis functions to be evaluated is on average approximately a factor of three less than the number of the features.

To incorporate the idea of features for the LSST scheduling problem with different requirements for different sky regions, we define a set of common basis functions and then modify them according to the requirements of each region.

4.2.1. Common basis functions

Generally speaking, making a decision for a visit at t for the LSST scheduling problem is mainly determined by the following factors,

1. The amount of the time it takes to redirect the telescope and the dome to move from one target to the next target.
2. The short-term science-driven requirements, such as the same-night revisit of a field.
3. The long-term mission-driven requirements, such as maintaining a uniform coverage of all field-filter pairs within each region.
4. The relative quality of visiting a field-filter compared to other field-filters at the time of observation, for overall efficiency of the operation.
5. The general preference for observing the fields around the meridian.

Accordingly, the basis functions of the LSST scheduler, are designed to formalize the above factors. Common basis functions which are shared amongst all of the regions, are defined in Table 2. Note that In the definition of Φ_1 , scale factor $s_1 = 0.43$ is empirically evaluated to ensure that 80% of the observed values of this basis function, for the visited states, in the simulations are between 0 and 1. Without loss of generality, scaling the values of the basis functions is a regulation that improves the rate of training convergence with a uniform numerical scale of the solution path.

Table 2. Basis functions of the Feature-Based scheduler, the building blocks of the decision function.

Notation	Definition\Description
$\Phi_1(f_1(i, f, t))$	$s_1 \cdot f_1(i, f, t)$, the cost of the required time for visiting field-filter (i, f) .
$\Phi_2(f_2(i, f, t))$	$\begin{cases} 0.5, & \text{if } \sum_f f_2(i, f, t) = 0 \\ 1, & \text{if } \sum_f f_2(i, f, t) = 2, \\ 0, & \text{else,} \end{cases}$ reflects the short term visit/revisit priority of field i , conditioned on the total number of the previous same-night visits.
$\Phi_3(f_5(i, t))$	$(1 - \frac{f_5(i, f, t)}{\max_{\ell} \max_{\phi} f_4(\ell, \phi, t)})$, reflects the long-term visit priority of field-filter (i, f) , based on the ratio of its co-added depth to the maximum co-added depth of all pairs of field-filter until t .
$\Phi_4(f_6(i, f, t))$	$1 - Pr(f_6(\ell, \phi, t) \leq f_6(i, f, t))$, empirical complimentary CDF of 5σ -depth of all (i, f) pairs at t . Φ_4 assigns a cost to field-filter (i, f) based on its relative visiting quality compared to the other field-filter pairs at t .
$\Phi_5(f_6(i, f, t))$	$\frac{ hr(i, t) }{12}$, encourages visiting of the fields near the meridian.

4.2.2. Region-specific modifications of the common basis functions

As mentioned in Section 4, the LSST’s mission poses different requirements on different regions of the sky. First we modify Φ_2 , for the Wide Fast Deep (WFD) and North Ecliptic Spur (NES) regions. Because they require the telescope to observe a field twice at a same night, and within a valid time window, $[W_1, W_2]$. The following modification prioritizes the fields that have received a first visit, but not a second visit.

$$\Phi_2^{\text{WFD}}(f_2, f_3, f_4) = \begin{cases} \Phi^{\text{pair}}(f_3) I_{f_4 > W_2}, & \text{if } \sum_f f_2 = 1, \\ \Phi_2(f_2), & \text{else,} \end{cases}$$

where, $\Phi^{\text{pair}}(f_3) = \exp(-\frac{\min_{\phi} f_3(i, \phi, t)}{W_2})$, if $\min_{\phi} f_3(i, \phi, t) < \frac{W_2}{2}$, to rank the fields that have received their first visit of the night (hence the condition). For any other cases, $\Phi^{\text{pair}}(f_3)$ is zero.

For the Deep Drilling Field (DDF) survey which contains a very small fraction of the visible sky’s area, instead of adjusting the basis functions, we prescribe a schedule, and interrupt the operation of the scheduler. Fortunately, the recoverability attribute of the Feature-Based scheduler enables the scheme of the interruptions to be a part of the decision making procedure as long as the interruptions are not too frequent.

4.2.3. Controllability of the scheduler

As discussed in Section 2, the mission’s objective has to be controllable via the design parameters, θ . In this section we present an empirical observation of the mission’s objective controllability. If the value of the objective function does not sufficiently respond to the variations of the design parameters, it is a sign of poor choice of the basis functions and/or input information. Either of these factors lead to a structure that does not admit a sufficiently optimal solution, and if on the other hand, the objective function is extremely variable with respect to the changes in the design parameters, the solution of the training is not reliable because the objective is not a well-behaved function of the optimization variables.

Fig 2, shows a few one dimensional slices of the two following simple objective functions (14) and (15), evaluated after about five hours of scheduling simulation, with different θ ’s in $[0, 10]$ range. To observe

the variability of the objective function with changes in the i 'th basis function, we define a sequence of equidistant values for $\theta_i \in \{\theta_i^{j_1}, \dots, \theta_i^{j_m}\}$ and keep the other θ 's fixed, then run the scheduler for $T = 4.8\text{hrs}$ with all resulting θ 's that are different by the i 'th element, then evaluate U_1 , and U_2 , defined as follows. U_1 reflects the slew time $\text{slew}(\cdot)$ and airmass $\text{am}(\cdot)$ aversion, and U_2 reflects the time-efficiency of the operation.

$$U_1(x_0, x_1, \dots, x_n) = - \sum_{\{i: t_0 < t^i < t_n\}} \text{slew}(id(t_i), id(t_i)) + 10.\text{am}(id(t_i)) \quad (14)$$

$$U_2(x_0, x_1, \dots, x_n) = n \quad (15)$$

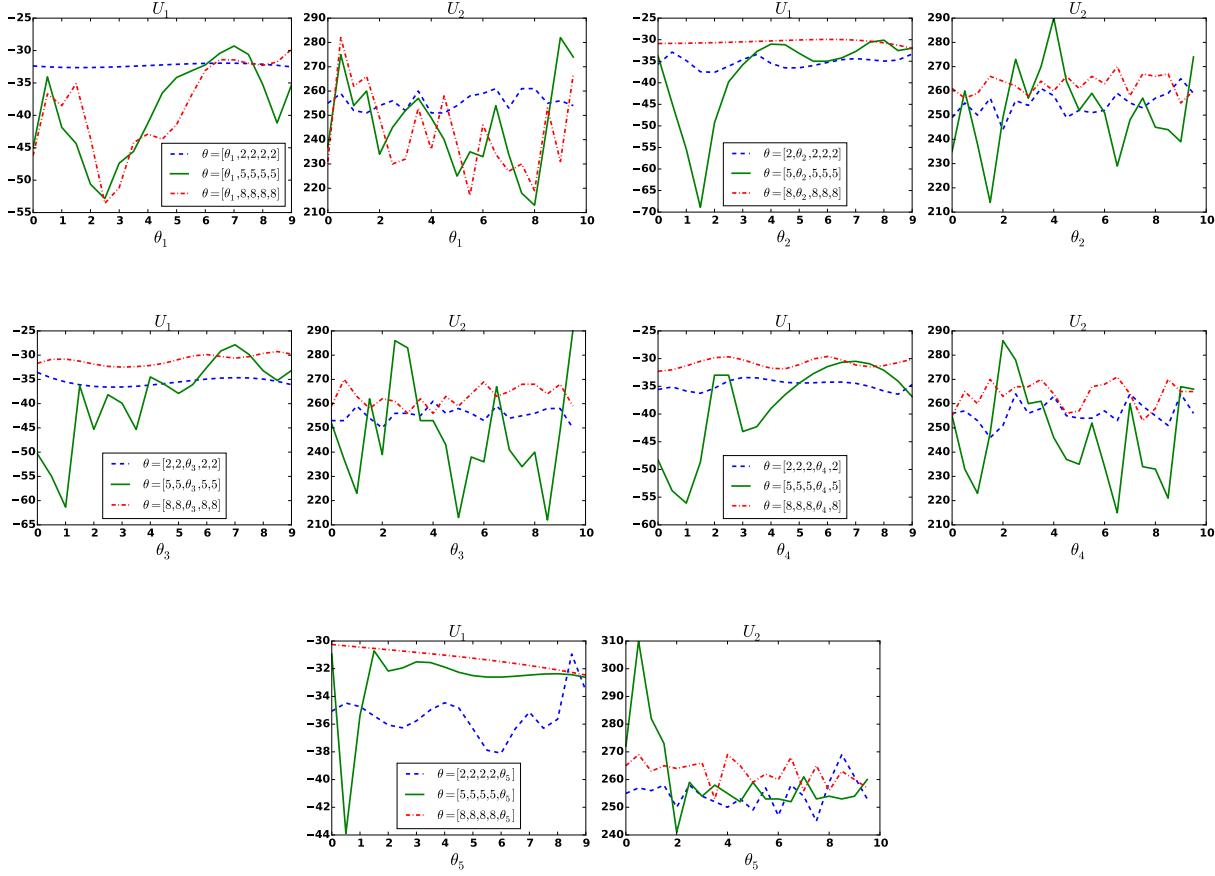


Figure 2. One dimensional slices of the two simple objective functions (14) and (15). The variation of the objective functions, specially in the mid-range slices (solid line) suggests that the scheduler's performance is controllable with the design parameters $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ within the proposed range of the search. This is of course valid for the performance that is measured with either of U_1 , or U_2 .

Figure 2, contains slices of the 5-dimensional U_1 and U_2 . Both of the simple objective functions reasonably respond to the changes in all five dimensions of the variable θ , which can be an evidence of the controllability of the objective function. Moreover, the smaller variations for slices closer to the

Table 3. Feasibility of field-filter (i, f) for a visit at t^{n+1} , as evaluated at t^n .

	Constraints	Description	region	
1	$\tau_{rise}(i, f, t^{n+1}) \leq t^{n+1} \leq \tau_{set}(i, f, t^{n+1})$	field-filter (i, f) has to be above the acceptable airmass horizon at t^{n+1} .	All regions	
2	$E[f_4(i, f, t^{n+1})] \neq 0$	field-filter (i, f) is not temporarily masked at t^{n+1} .	All regions	
3	$\sum_f f_2(i, f, t^n) < N^{survey}$	N^{survey} poses a region dependent upper-bound on the number of the visits for each field. $N^{WFD} = N^{NES} = 3$, and $N^{GPR} = N^{SCP} = 1$.	All regions	
4	$E[f_6(i, f, t^{n+1})] < \sigma(survey, f)$	the expected quality of visiting field-filter (i, f) at t^{n+1} has to be better than the given threshold, $\sigma(\cdot)$, that depends on the survey and the filter.	All regions	
5	$f \neq id(t^n)$	consecutive visit of a same field is not allowed.	All regions	
6	if $\sum_f f_2(i, f, t^n) = 0$ then $\max_\phi f_4(i, \phi, t^n) > \frac{W_1 + W_2}{2}$	the first visit of field f has to occur $\frac{W_1 + W_2}{2}$ time before it becomes invisible, so that the second visit of f can be scheduled in the valid time window.	WFD and NES	
7	if $\max_\phi \theta(i, \phi, t^n) > \tau_s(t^n)$ then $W_1 \leq \min_\phi f_2(i, \phi, t^n) \leq W_2$	if there has been a same-night visit of field f until t^n , then the next same-night visit has to occur in the valid time window.	WFD and NES	
8	if $\max_\phi \theta(i, \phi, t^n) > \tau_s(t^n)$ then $f \notin \{y, u\}$	if there is a same-night visit of field f until t^n , then the next same-night visit cannot be with either of u or y filters.	WFD	
9	$f \notin \{y, u\}$	visits with u filter and y filter is not allowed.	NES	

boundaries of the search space, suggest that the design and scaling of the basis functions provide a desirable behavior within the proposed search space.

4.3. Survey-specific constraints

The scheduler's decision at each time step is an admissible (feasible and measurable) pair of field-filter (i, f) , thus before each decision, one needs to specify the set of feasible actions. Feasibility of a candidate pair is driven by the following measurable factors:

- Visibility: The candidate field-filter has to be visible.
- Quality: The expected observational quality of a field-filter has to be beyond the given lower threshold.
- Survey's timing: The science-Driven revisit constraints has to be respected.

Exact expression of the proposed constraints for the LSST scheduler are presented in Table 3.

4.4. Scheduler optimization

For both of the reinforcement learning and the global optimization approaches, discussed in Section 3 we used a simulated model of the telescope and the environment, including the brightness of the sky and coverage of the clouds which are developed based on the measurements at the LSST site.

The simulation for the reinforcement learning starts at $t_0 = 2462867.5 \text{ mjd}$ (January 1, 2021), with $\theta^0 = (5, 5, 5, 5, 5)$, initialized at the mid-range values, and continues until θ converges. Figure 3, is the training curve for all of the variables over a course of 3000 decisions, for the instant reward of $r(s_{i-1}, s_i) = -\text{slew}(id(t_{i-1}), id(t_i)) - \text{am}(id(t_i))$. The discount rate $\gamma = 0.9$, and learning rate $\frac{0.01}{\log^3(i)}$ are chosen empirically.

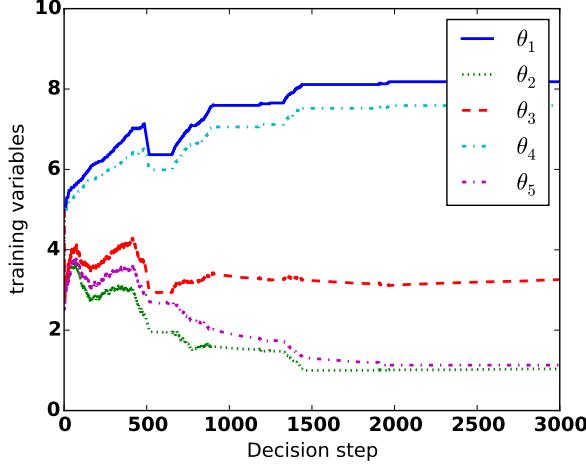


Figure 3. Reinforcement learning of the scheduler’s parameters, $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. All of the parameters are initialized at the mid-range value, 5. During the simulation, at each step there is a reward associated with the decision which implies a small adjustment on each of the five parameters. Then the next decision will be taken with a slightly different set of variables. This procedure continues until the adjustments on the variables are negligible.

For the above choices of reward, learning rate, discount factor and initialization, θ converges to $\theta^* = (8.18, 1.04, 3.26, 7.59, 1.13)$. In this approach, the computational time to evaluate the update is negligible with respect to the time required to evaluate the features, therefore, the computational time of the reinforcement learning is almost equal to the scheduling simulation (without the updates on θ) which is linear with respect to the number of the decisions. With a personal computer², each decision takes about 0.8 second, thus the time of the convergence for the simulation, presented in this section is $3000 \times 0.8 \text{ sec} = 40 \text{ minutes}$.

One of the important simple objective functions that cannot be expressed as the discounted sum of the rewards is the total number of the observations from any t_i to any t_j that can be expressed as a utility function, $U_{\pi_\theta}(x_i, x_{i+1}, \dots, x_j) = j - i$.

To find a set of parameters, θ , that optimize the above objective function we applied the global optimization approach, explained in Section 3, with the following regulatory constraints.

- (1) $\theta \geq 0$: Positive coefficients for the basis functions are assumed in the design of the basis functions.

² Processor: 1.6 GHz, Memory 1600 MHz DDR3

Because in the context of the telescope scheduling, it is more natural to create the basis functions to reflect the cost of the operation. (2) $\theta_1 = \theta_0$: Without loss of generality, we fix the value of the first element of θ to reduce the dimension of the optimization problem by one. Because, homogeneity of the policy implies that if θ^* yields an optimal scheduler, then $\alpha\theta^*$ for $\alpha > 0$ yields an optimal scheduler too.

We used the above objective function, U_{π_θ} , for $t_i = 2462867.5$ *mjd* (January 1, 2021) and $t_j = 2462877.5$ *mjd* (January 11, 2021). Figure 4, shows the value of this objective function over the iterations of the *eDE* algorithm. The solution $\theta^* = (1.00, 0.84, 0.99, 1.34, 3.04)$, yields the best U_{π_θ} after 50 iterations for $\theta_1 = 1.00$

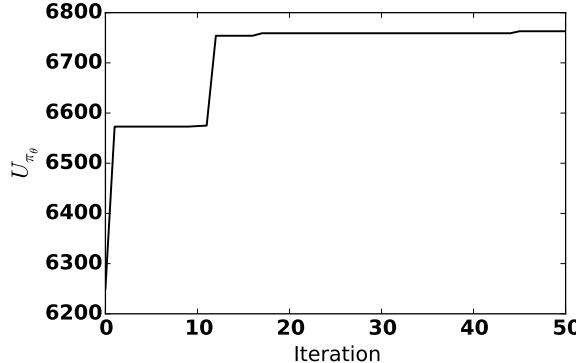


Figure 4. Progress of the black-box objective function U_{π_θ} over the iterations of the *eDE* algorithm. U_{π_θ} , for this simulation is the total number of the observations for 10 nights starting from January 1, 2021.

eDE is a population-based metaheuristic algorithm, and for the result shown in Figure 4, the number of population N_P is set to be 50. Each function evaluation is in fact, the simulation of 10 days of scheduling with a candidate scheduler which takes about 8 minutes, therefore each iteration, in total, takes $N_P * 8$ minutes with a personal computer. The optimization can be manually terminated if the result is satisfactory, or can be continued until a full convergence is achieved. In *eDE* (and all genetic algorithms in general), function evaluation for each individual is independent from other individuals, therefore the parallel implementation of the same algorithm can be faster up to a factor of N_p .

5. PERFORMANCE OF A MODIFIED FEATURE-BASED SCHEDULER FOR LSST

In this section, the LSST Metric Analysis Framework (Jones et al. 2014) is used, to compare³ the performance of a modified version of the Feature-Based scheduler with *astro-lsst-01-2013*, the most recent official schedule of the LSST, over a 10-year period of scheduling simulations.

The *Modified Feature-Based* scheduler is under active development⁴, and addresses the observational details of the LSST’s mission through the adjustment of the constraints and the basis functions. It is designed to produce a software that can be used in practice. In this version, we adopt a finer discretization of the sky, and do not require the partitions to be necessarily sized based on the telescope’s field of view. The fact that the policy is not computationally expensive to evaluate makes it possible to use a finer discretization of the feature-space. This approach allows the scheduler to handle the field’s overlaps

³ The sky background models and weather downtime used to benchmark the algorithms are not exactly identical because of the practical difficulties in the separation of the environment and the scheduler in the *astro-lsst-01-2013*’s implementation. However, for the purpose of the comparisons in this paper, the behavior of our sky and observatory model is sufficiently close to the official model.

⁴ GitHub repository: https://github.com/lsst/sims_featureScheduler.

which cause inhomogeneity in the coverage of the sky. In addition to adopt a finer discretization, we use a spatial dithering scheme to randomize the final pointing of the telescope by a small amount around the center of the partitions to further assist the homogeneity of the coverage. Moreover, the Modified Feature-Based scheduler uses a separate process to track if an observation will need to be observed in a pair, and a separate processes to decide if a Deep Drilling sequence should be executed by interrupting the normal operation of the telescope.

For a survey telescope, such as the LSST, the density of the co-added depth over the visible sky has to be ideally uniform in each filter and within each of the five survey regions. Figure 5 compares the values of the co-added depth on a discretized sky map. Figure 6, demonstrates the smoothness of the coverage in a smaller scale for *astro-lsst-01-2013*, with and without dithering, and compares it with that of the Modified Feature-Based scheduler. Figure 6 are zoomed-in versions of the co-added depth coverage in the *r* band, around the boundary of WFD and GP regions. However the granular coverage pattern of *astro-lsst-01-2013*, and smooth coverage of the Modified Feature-Based scheduler is similar for all of the filters. Smoother coverage that the Modified Feature-Based scheduler offers is due to the fine discretization of the sky in the decision making stage, in addition to dithering which is applied after the decision is made.

Figure 7 and Table 7 compare the distribution of the co-added depth on a (finely) discretized sky. The Modified feature-Based scheduler has paved the left-most peak that appears in the distribution of *astro-lsst-01-2013*. This peak is the result of the field's overlaps that receive more number of visits than it's assumed in the decision making procedure of the scheduling. The default sky tessellation adopted by LSST results in 23% of the sky being covered by more than one field. Adopting the dithering scheme, the median number of observations at a typical point in the sky increases by $\sim 15\%$. Dithering is also essential for removing systematic effects for science cases such as measuring galaxy counts, (see (Awan et al. 2016) for more details).

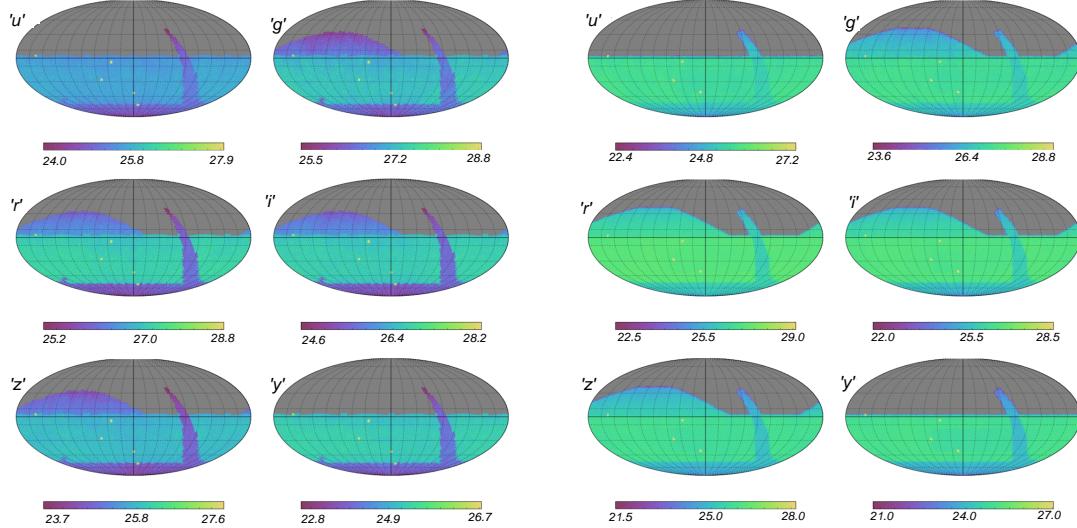


Figure 5. The final co-added depth coverage in each of the six filters $\{u, g, r, i, z, y\}$. According to the mission's objective, the scheduler has to provide a uniform coverage of the visible sky within each region, and in each filter. The left panels show the *astro-lsst-01-2013* simulation results while the Modified Feature-Based scheduler is on the right. Modified Feature-Based scheduler closely matches the footprint of the official survey.

In addition to uniformity of the coverage, the LSST mission calls for pairs of visits within a valid time window at the same night. The main reason is to detect the transient objects such as asteroids. Since,

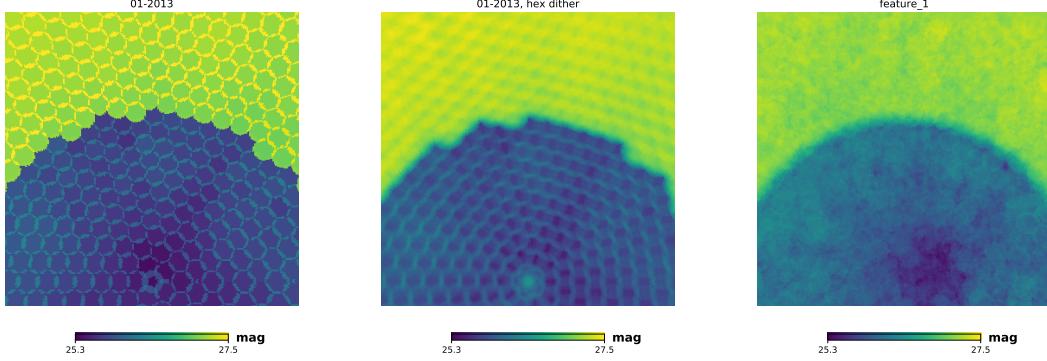


Figure 6. Smoothness of the co-added depth coverage in a smaller-scale, in the r band. The smooth coverage of the Modified Feature-Based scheduler (right) versus the granular pattern of *astro-lsst-01-2013* (left), further satisfies the uniformity of the coverage which is one of the most fundamental objectives of the LSST mission. The middle figure is the coverage of *astro-lsst-01-2013*, with dithering of the same sequence which fundamentally can not become as smooth as the right figure, because unlike Feature-Based scheduler, the scheme of *astro-lsst-01-2013* does not allow for decision making with arbitrary discretization of the sky

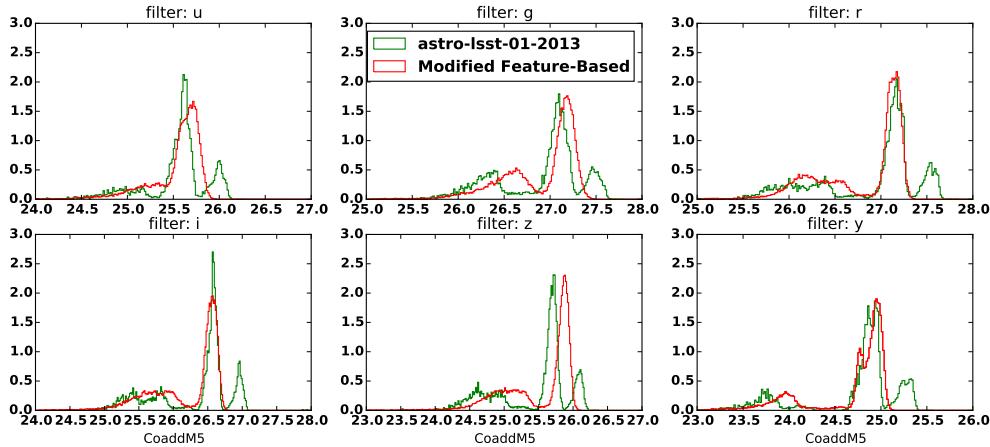


Figure 7. Each plot compares the distributions of the co-added depth coverage in one of the six filters. A dithering scheme in the Modified Feature-Based scheduler smoothens the depth of the coverage where the fields overlap. Consequently and in contrast with *astro-lsst-01-2013*, one cannot find an extra unwanted peak on the right extreme of the distribution. As a consequence of paving the overlap peak, the telescope's time is spent on improving the median of the depth coverage distribution that amounts to a higher co-added depth for a larger area of the sky.

the moving objects usually belong to the solar system, the pair constraint was initially imposed only on the WFD and NES regions. However, there are interesting solar system objects such as interstellar asteroids that can be observed in any direction of the sky, besides identification of the other varying objects, such as super novae, can benefit from a follow up visit, especially if the second visit is with a different filter. Thus in the Modified Feature-Based scheduler we made the pair constraint a universal constraint for all of the regions. The downside of this extension is the fact that it constrains the scheduler even more and the performance can be potentially less than it could be. Note that the structure of the

Table 4. The median and variance of the co-added depth distribution on a finely discretized sky. Modified Feature-Based scheduler closely matches the footprint of the official survey, and in addition outperforms *astro-lsst-01-2013* in terms of the uniformity of the coverage, with lower variances, specially in WFD and SCP regions.

filter	Median, Variance							
	<i>astro-lsst-01-2013</i>				Modified Feature-Based			
	WFD	GP	SCP	NES	WFD	GP	SCP	NES
<i>u</i>	25.63, 0.04	25.12, 0.11	24.91, 0.12	-	25.68, 0.01	25.32, 0.05	25.10, 0.04	-
<i>g</i>	27.13, 0.04	26.41, 0.09	26.32, 0.12	26.30, 0.13	27.18, 0.01	26.69, 0.04	26.56, 0.04	26.47, 0.09
<i>r</i>	27.19, 0.04	26.01, 0.16	25.84, 0.24	26.38, 0.12	27.14, 0.01	26.21, 0.07	26.08, 0.05	26.43, 0.09
<i>i</i>	26.60, 0.04	25.44, 0.15	25.28, 0.22	25.82, 0.12	26.56, 0.01	25.68, 0.08	25.43, 0.07	25.88, 0.09
<i>z</i>	25.73, 0.04	24.62, 0.17	24.57, 0.21	24.90, 0.14	25.87, 0.01	25.03, 0.9	24.81, 0.05	25.16, 0.10
<i>y</i>	24.92, 0.04	23.81, 0.16	23.72, 0.21	-	24.92, 0.02	24.01, 0.09	23.88, 0.06	-

Feature-Based scheduler, allows for extension or restriction of the constraints down to the individual field’s level, with neither contradicting any of the Markovian framework assumptions, nor breaking the structure of the implementation. Figure 8 demonstrates the distribution of the observations in pair (in the *g*, *r*, and *i* filters) to the total number of the observations. For the regions that the pair constraint is applied, this ratio can be interpreted as the success rate of the scheduler to satisfy the pair constraint. Figure 9, compares the distribution of the visits-in-pair ratio of the modified Feature-Based scheduler and *astro-lsst-01-2013*. Note that the peak of the density for Figure 9 is closer to 1, which means a larger area of the sky is covered by successful visits-in-pair, however, the Modified Feature-Based scheduler offers a sharper concentration of the values that can be interpreted as a more homogenous visits-in-pair ratio. In other words, the Modified Feature-Based scheduler, sacrifices perfect visit-in-pair for a limited area of the sky to maintain a uniform ratio of visit-in-pair for a larger area of the sky.

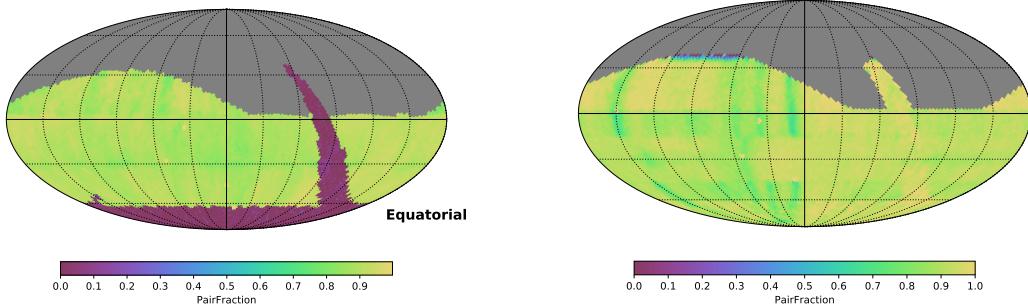


Figure 8. The ratio of the visit-in-pairs (in the *g*, *r*, and *i* filters) to the total number of the observations on the sky map. For the areas that the pair constraint is applied, the ratio is desired to be one. For Modified Feature-Based scheduler (right), the pair constraint is applied to all of the regions and for *astro-lsst-01-2013* (left), it is applied to the WFD and NES regions only.

For a ground-based instrument, airmass is one of the major obstacles for high-quality observation. Although zenith observations have the minimum airmass, off-zenith observations cannot be avoided, in which case, observations around the meridian provide high quality data and consequently result in more efficient operation of the telescope. Figure 10 compares the number density of the visits on an altitude-azimuth sky map in each of the six filters [*u, g, r, i, z, y*]. Clearly in all of the filters, the modified

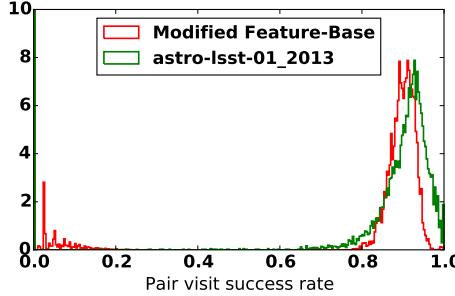


Figure 9. Distribution of the visits-in-pair ratio to the total number of the observations, in any of the g , r , and i filters. The Feature-Based scheduler in compare with *astro-lsst-01-2013*, sacrifices perfect visit-in-pair for a limited area of the visible sky (with a lower mean) to maintain a uniform ratio of visit-in-pair for a larger area of the sky (with a sharper distribution)

Feature-Based scheduler schedules more visits around the desirable meridian zone. In addition, it offers a consistent concentration peak on the east wings, which is essential for a higher success rate for the visit-in-pair constraint. Because, if the first visit of the night occurs when the field is on the east side of the sky, it provides a longer time opportunity for the second visit of the same night. Note that, adjustability of the Feature-Based scheduler allows for a significant change in the behavior, in this case a variant of the marching army cadence is incorporated to control the location of the visits through a basis function modification.

For a multi-mission, survey telescope, such as LSST, comparing the overall performance of the different schedules is a difficult task. Particularly because of the large number of the competing factors that are involved in the performance evaluation. In some cases involving the importance of each area of astronomy, the criteria are not even objective. Nevertheless we conclude this section with a general comparison by combining the surveys' open shutter fraction (OSF), and median airmass of the observations. The open shutter fraction is the total time that the telescope camera shutter was open divided by the total time it could have been open. This reveals how time-efficiently the observations have been scheduled. The median airmass reflects the overall quality of the collected data. As mentioned before, observations in lower airmass allows for a higher data quality. Figure 11 visualizes the number density of the visits on an altitude-azimuth sky map for the modified Feature-Based scheduler with the most recent LSST baseline schedules *astro-lsst-01-2013* and *minion_1016*. Table 9, shows that there is a trade-off between airmass and OSF. For instance *minion_1016*, offers the best OSF (time efficiency), while the median airmass is the highest. (data quality). Meanwhile *astro-lsst-01_2013* offers low airmass observations, resulting in a lower OSF. Moreover, in the r filter where atmospheric extinction is less relevant, *minion_1016* has a higher total throughput while *astro-lsst-01_2013* has higher throughput in the g filter. By including dithering, the Modified Feature-Based scheduler vastly outperforms the other surveys in both the blue and red filters. The Modified Feature-Based simulation has the lowest OSF, but this is an initial simulation where the basis function weights have not been optimized, so there is potential for further improvements.

6. CONCLUDING REMARKS

This study demonstrated that a Markovian scheduler based on expert-designed features, and a parametrized decision making policy can be successfully applied to multi-mission, ground-based telescopes such as LSST. Unlike the mainstream telescope schedulers, Feature-Based scheduler does not rely on hand-crafted observation proposals. Instead, by bringing the decision making process to the individual observation level, improves the efficiency of the telescope's operation. In particular, because with this approach the telescope's schedule can be designed for optimality in addition to feasibility, and in general,

Table 5. Comparison of three different survey algorithms in a section of the LSST Wide-Fast-Deep survey area.

Survey	OSF	median Airmass	dithered	median throughput	r (%)	g (%)
Modified F-B	0.705	1.1	yes	63.7	47.0	
minion_1016	0.736	1.2	no	55.3	40.0	
astro-lsst-01_2013	0.715	1.1	no	54.4	40.8	

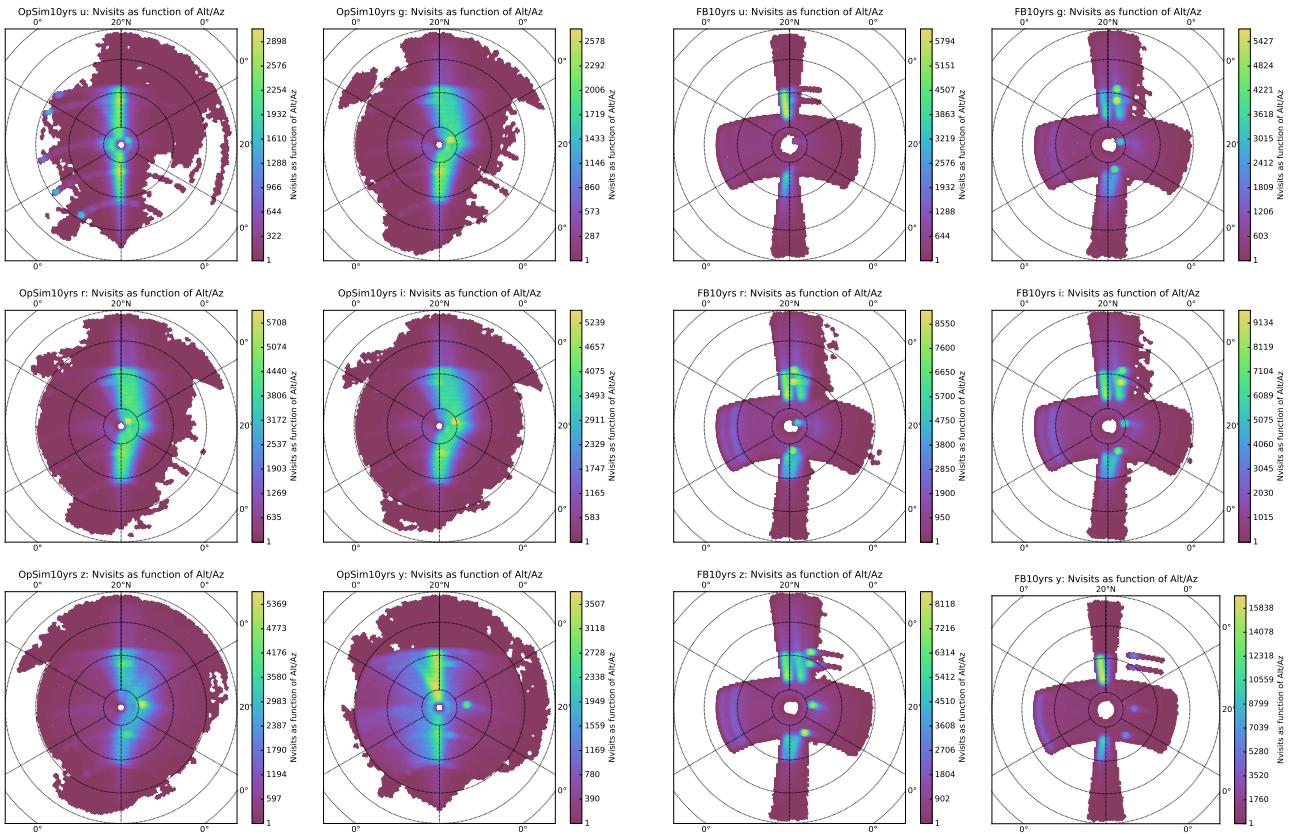


Figure 10. Each plot is the distribution of the number of the visits on an altitude-azimuth sky map in one of the six filters. The two left columns belong to *astro-lsst-01-2013*, and the two right columns belong to a simulation of the Modified Feature-Based scheduler. The higher concentration on the meridian (vertical axis) for the Modified feature-Based scheduler shows a more desirable behavior. Moreover, consistent concentration of the visits on the east wing can potentially provide a better visit-in-pair behavior

because schedulers which rely on human interaction are fundamentally prone to potential suboptimality, mainly due to the manual tailoring which is performed based on the inspections of the instances. Moreover, adjusting their behavior is inconvenient and time consuming in practice. Furthermore, being

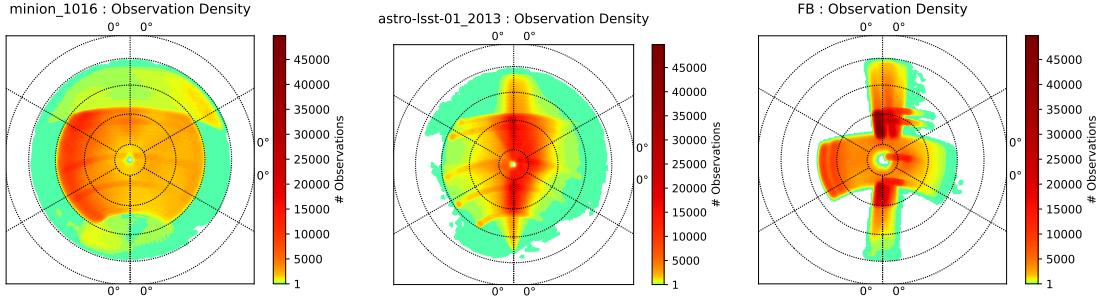


Figure 11. The distribution of the number of the visits on an altitude-azimuth sky map for three schedulers. On the left, the minion_1016 simulation schedules most observations at high airmass and very few on the meridian. In the middle, the astro-lsst-01_2013 simulation schedules many observations on the meridian, but still executes deep drilling fields and a smattering of other observations at high airmass. On the right, the Modified Feature-Based scheduler concentrates observations around the meridian, including deep drilling observation.

modeled as a Markovian Decision Process, the Feature-Based scheduler offers a systematic approach to the optimization of the scheduler’s behavior under uncertainties and interruptions.

On the other hand, the decision elements in the Feature-Based scheduler are designed, and separated in an intuitive way for the astronomy community. This property allows for expert intervention if needed, however in a regulated way, and only on the parameters of the policy. Manual adjustments of the parameters of the policy does not break measurability, linearity, and memorylessness of the process. Thus all of the simplicity, optimality conditions and modularity of the design remain valid. In addition, due to the coherent structure, from training to online decision-making, the Feature-Based scheduler is easy to understand, implement, and troubleshoot. Simplicity of the design and implementation, also provides a desirable environment for a wide-range of programming expertise in astronomy community to install the python packages on a local computer, define a custom mission objective, train a scheduler and examine the behavior of the scheduler with various mission objectives. Similarly, in a particular project, when a change in the mission’s objective is necessary, deriving a new scheduler that optimizes the new objective is principally automated. Furthermore, for the mission planning stage of a future instrument, a scheduler with adjustable objective can be extremely helpful, because it can answer the high-level trade-off questions, such as the amount of the time efficiency loss in the different strategies of capturing transient objects.

Computationally, the required resources for the training/optimizing of the Feature-Based scheduler is versatile, depending on the purpose. If many different objective functions are being tested for planning a mission, then a quick eDE optimization for a short scheduling episode can find a sufficiently good scheduler for each mission. Even a quick manual hand tuning that reflects the intuitive importance of each basis function is possible, because they carry an astronomical observation meaning. On the other hand if the objective is known and fixed, and the scheduler is being trained for real-time decision making, then one might even categorize the observation nights based on their main differences, such as the moon-phase, and train a scheduler specifically for each category.

Acknowledgments. We would like to thank the LSST corporation’s leadership for the financial support of this work, and the DIRAC Institute’s faculty and researchers for providing expertise, without which this research would not have been possible. In particular, Professor Željko Ivezić whose consistent attention and insightful comments greatly assisted this work. We would also like to show our gratitude to the experts in other institutes, specially Professor Michael Strauss, Professor Christopher Stubbs, Dr.

Robert Lupton, and Michael Reuter for their insightful comments spanning from the idea stage to the end of this project.

Facility: LSST

Software: MAF (Jones et al. 2014), healpy (Górski et al. 2005), matplotlib (Hunter 2007), astropy (The Astropy Collaboration et al. 2018), numpy/scipy (Jones et al. 2001)

REFERENCES

- Alexander, K. D., Berger, E., Bower, G., et al. 2017, ArXiv e-prints. <https://arxiv.org/abs/1703.04692>
- Awan, H., Gawiser, E., Kurczynski, P., et al. 2016, ApJ, 829, 50, doi: [10.3847/0004-637X/829/1/50](https://doi.org/10.3847/0004-637X/829/1/50)
- Bellm, E. 2014, in The Third Hot-wiring the Transient Universe Workshop, ed. P. R. Wozniak, M. J. Graham, A. A. Mahabal, & R. Seaman, 27–33
- Boroson, T., Brown, T., Hjelstrom, A., et al. 2014, in Proc. SPIE, Vol. 9149, Observatory Operations: Strategies, Processes, and Systems V, 91491E
- Brink, J., Charles, A., Hettlage, C., et al. 2008, in Proc. SPIE, Vol. 7019, Advanced Software and Control for Astronomy II, 70190N
- Delgado, F., & Reuter, M. A. 2016, in SPIE Astronomical Telescopes+ Instrumentation, International Society for Optics and Photonics, 991019–991019
- Delgado, F., Saha, A., Chandrasekharan, S., et al. 2014, in Proc. SPIE, Vol. 9150, Modeling, Systems Engineering, and Project Management for Astronomy VI, 915015
- Garcia-Piquer, A., Ribas, I., & Colomé, J. 2015, Experimental Astronomy, 40, 671
- Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, ApJ, 622, 759, doi: [10.1086/427976](https://doi.org/10.1086/427976)
- Gurobi Optimization, I. 2016, Gurobi Optimizer Reference Manual. <http://www.gurobi.com>
- Hunter, J. D. 2007, Computing In Science & Engineering, 9, 90, doi: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
- Ivezic, Z., Tyson, J. A., Abel, B., et al. 2008, ArXiv e-prints. <https://arxiv.org/abs/0805.2366>
- Ivezic, Z., Tyson, J., Abel, B., et al. 2008, arXiv preprint arXiv:0805.2366
- Johnston, M. D., & Miller, G. 1994, Intelligent Scheduling, 391
- Jones, E., Oliphant, T., Peterson, P., et al. 2001, SciPy: Open source scientific tools for Python. <http://www.scipy.org/>
- Jones, R. L., Yoachim, P., Chandrasekharan, S., et al. 2014, in SPIE Astronomical Telescopes+ Instrumentation, International Society for Optics and Photonics, 91490B–91490B
- Jones, R. L., Slater, C. T., Moeyens, J., et al. 2018, Icarus, 303, 181, doi: [10.1016/j.icarus.2017.11.033](https://doi.org/10.1016/j.icarus.2017.11.033)
- Lampoudi, S., Saunders, E., & Eastman, J. 2015, ArXiv e-prints. <https://arxiv.org/abs/1503.07170>
- Muscettola, N., Pell, B., Hansson, O., & Mohan, S. 1995, in Robotic Telescopes. Current Capabilities, Present Developments, and Future Prospects for Automated Astronomy, Vol. 79, 148
- Naghib, E., & Nobakhti, A. 2016, in American Control Conference (ACC), 2016, IEEE, 2440–2446
- Naghib, E., Vanderbei, R. J., & Stubbs, C. 2016, in Proc. SPIE, Vol. 9910, Observatory Operations: Strategies, Processes, and Systems VI, 991011
- Nelson, J. E., Mast, T. S., & Faber, S. M. 1985
- Nowakowski, J., Schwärzler, W., & Triesch, E. 1999, European Journal of Operational Research, 112, 531
- Saunders, E., & Lampoudi, S. 2014, in The Third Hot-wiring the Transient Universe Workshop, ed. P. R. Wozniak, M. J. Graham, A. A. Mahabal, & R. Seaman, 117–123
- Sebag, J., Barr, J., Saha, A., Claver, C., & Krabbendam, V. 2007, in Revista Mexicana de Astronomia y Astrofisica Conference Series, Vol. 31, Revista Mexicana de Astronomia y Astrofisica Conference Series, 30–35
- Sebag, J., Krabbendam, V. L., Claver, C. F., et al. 2008, in Proc. SPIE, Vol. 7012, Ground-based and Airborne Telescopes II, 70123W

- Shetrone, M., Cornell, M. E., Fowler, J. R., et al. 2007, Publications of the Astronomical Society of the Pacific, 119, 556
- Steele, I. A., & Carter, D. 1997, in Proc. SPIE, Vol. 3112, Telescope Control Systems II, ed. H. Lewis, 222–233
- Storn, R., & Price, K. 1997, Journal of global optimization, 11, 341
- Tesauro, G. 1995, Communications of the ACM, 38, 58
- The Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, ArXiv e-prints. <https://arxiv.org/abs/1801.02634>
- Tsitsiklis, J. N., & Van Roy, B. 1997, in Advances in neural information processing systems, 1075–1081
- Wootten, A. 2003, in Proc. SPIE, Vol. 4837, Large Ground-based Telescopes, ed. J. M. Oschmann & L. M. Stepp, 110–118
- Yoachim, P., Coughlin, M., Angeli, G. Z., et al. 2016, in Proc. SPIE, Vol. 9910, Observatory Operations: Strategies, Processes, and Systems VI, 99101A