

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

ELAHESADAT NAGHIB,<sup>1</sup> PETER YOACHIM,<sup>2</sup> ROBERT J. VANDERBEI,<sup>1</sup> ANDREW J. CONNOLLY,<sup>2</sup> AND R. LYNNE JONES<sup>2</sup>

<sup>1</sup>*Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08540, USA*

<sup>2</sup>*Department of Astronomy, University of Washington, Box 351580, U.W., Seattle, WA 98195, USA*

### Abstract

**How ground-based telescopes schedule their observations in response to different variables can significantly impact their efficiency. Namely, competing science priorities and constraints, variations in the weather, and the visibility of a particular part of the sky are some of the determining variables.** In this paper we introduce the Feature-Based telescope scheduler which is an automated, proposal-free decision making algorithm. It offers *controllability* of the behavior, *adjustability* of the mission, and quick *recoverability* from interruptions for large ground-based telescopes. By framing Feature-Based telescope scheduler in the context of a coherent mathematical model the functionality and performance of the algorithm is simple to interpret. Consequently adapting this scheduler for a broad range of ground-based instruments is straightforward. This paper presents a generic version of the Feature-Based scheduler, with minimal manual tailoring. We demonstrate its potential and flexibility to serve as a foundation for schedulers of the ground-based instruments. In addition, a modified version of the Feature-Based scheduler for the Large Synoptic Survey Telescope (LSST) is introduced and compared to the previous LSST schedulers.

**Keywords:** Artificial intelligence, autonomous telescope, LSST, reinforcement learning, scheduling, stochastic optimization

### 1. INTRODUCTION

The Large Synoptic Survey Telescope (LSST), is a large, ground-based optical survey that will image half of the sky every few nights from Cerro Pachon in Northern Chile. The LSST comprises an 8.4 meter primary mirror and a 3.2 Gigapixel camera. With a 9.6 degree<sup>2</sup> field-of-view. It will visit each part of its 18000 degree<sup>2</sup> primary survey area about 1000 times over the course of 10 years. Each visit will likely comprise a 15 second pair of exposures with a single visit depth of about 24.5 magnitudes (AB) (in the six bands u, g, r, i, z, and y). The revolutionary role of this telescope calls for no less than optimal operation.

[enaghlib@princeton.edu](mailto:enaghlib@princeton.edu)

[yoachim@uw.edu](mailto:yoachim@uw.edu)

[rvdb@princeton.edu](mailto:rvdb@princeton.edu)

[ajc26@uw.edu](mailto:ajc26@uw.edu)

[ljones@astro.washington.edu](mailto:ljones@astro.washington.edu)

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.

There are four primary science drivers for the LSST project: the characterization of dark energy through the multiple cosmological probes (e.g. gravitational weak lensing, luminosity distances from Type Ia supernovae, and Baryon Acoustic Oscillations); mapping the 3D distribution of stars within our Galaxy; a census of solar system objects within the Solar System; and a detailed study of the transient and variable universe. Each of these objectives has a different set of constraints and requirements on how the observations are made (e.g. the cadence of the observations, the number of filters as a function of time, the acceptable airmass range for an observation). Generally, 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. 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. 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 pioneered many of the developments in algorithmic scheduling of observations for the space missions.

Despite the similarity of the science objectives for space and ground-based telescopes, the determining factors for the purpose of scheduling are fundamentally different. While space telescopes are required to respect kinematical and dynamical constraints, weather 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 computational limitations.

Earlier algorithmic approaches to the scheduling of ground-based telescopes are heavily based on observation *proposals*. Proposals are hand-crafted sequences of scripted astronomical observations. They are generally tested only for feasibility (e.g. that a set of fields were visible, or lie within a specified airmass range, or within a window in time), but not necessarily for optimality. For instance, the operation of Keck Telescope, 1993 ([Nelson et al. 1985](#)), is 100% based on proposals, and the Hobby-Eberly Telescope, 1997 ([Shetrone et al. 2007](#)), has a semi-manual scheduling scheme.

More recently, the development of more expensive ground-based instruments with complex missions made it impossible to rely solely on hand-crafted proposals. The need for more efficient use of the instrument's time led to the development of decision making algorithms to optimize their science output. **For instance, the scheduler of the Liverpool robotic telescope was designed in 1997 to automatically allocate time slots to chunks of scripted observations. The time allocation strategy was preferred to the scheduling at the single visit level. The scheduling at the single visit level is referred to as optimal scheduling by the authors ([Steele & Carter 1997](#)) where it is stated that the optimal scheduling requires to reevaluate the future sequence of observations once it is interrupted, but the necessary extra computation is neither affordable nor fast enough.** However, in this paper, we show that the scheduling in the single visit level, optimal scheduling, can be quickly recovered after an interruption, if a memoryless framework is used. Thus the optimality is not necessarily needed to be sacrificed because of the limited computational resources. 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 scheduler<sup>1</sup>. 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 (when there's a filter change), 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 scripted 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 scripted schedule is principally the lack of recoverability from unpredictable intrusions, such as inability to dodge the clouds.

Even given the reliability of fully automatic scheduling technologies, remain a number of modern telescopes such as SALT, 2005 (Brink et al. 2008), and ALMA, 2013 (Wootten 2003), which are being operated based only on 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.

The LSST community, however, in addition to a proposal-based scheduler, introduced in (Delgado & Reuter 2016), supported the design and implementation of proposal-free decision algorithms, such as the Feature-Based scheduler, first introduced in (Naghib et al. 2016), as well as a semi-scripted cadence by D. Rothchild et al. (2018, in preparation).

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 ground-based telescope with multiple science objectives, such as LSST, the scheduler has to offer *controllability*, *adjustability*, and *recoverability*.

- **Controllability:** A telescope is controllable by a given scheduler if change of the scheduler's design parameters are visibly manifested in the behavior and performance of the telescope. Controllability is determined by two factors, the information that is fed to the scheduler and the structure of the scheduler. Feeding the scheduler with information that is not sufficiently relevant to the high-level mission objectives leads to an output schedule that is irrelevant to the performance of the telescope. In addition to provide relevant information, a scheduler with a flexible structure is needed to cover a large class of decision strategies, each determined by a set of its design parameters. **Controllability is necessary for a scheduler to be optimizable by searching within the space of its design parameters.**
- **Adjustability:** A scheduler must be adjustable according to new conditions, environment and scientific desirables. For a complex and multi-objective mission it is common that the scientific goals are required to be modified in the middle of the operational period. Another example is aging that causes changes in the mechanical characteristics of the telescope. Regardless of the reason, **A scheduler is adjustable if the changes can be accommodated with reasonable**

<sup>1</sup> GitHub repository: [https://github.com/ZwickyTransientFacility/ztf\\_sim](https://github.com/ZwickyTransientFacility/ztf_sim)

**computational cost, and preferably no or minimal human-expert intervention. Hand-tuned scheduling strategies and policies that are written in forms of instructions for human observation proposal writers, 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 downtime and instrument failures are examples of the many unpredictable survey interruptions. In addition, there are inherently predictable interruptions, such as maintenance downtimes, and cable winding that due to the complexity of the mechanical system are not computationally affordable and/or valuable to keep track of. Therefore, they are considered to be stochastic variables as well. A scheduler is required to be able to make alternative decisions once a previously unpredictable event occurred in order to maintain the optimality of the operation. Moreover, it has to return to its normal behavior shortly after the interrupting event is over. **Such a scheduler is called recoverable and its response to the interruptions is ideally as quick as the length of a single observation. The response time is usually limited by computational complexity of the scheduling algorithm.** For instance, strategies that need to look back at history of observations or look forward through possible sequences of observations are not fast recoverables due to their computational complexity. Also, scripted sequences of observations, such as a proposals, completely lack the attribute of recoverability.

The Feature-Based scheduler is verifiably controllable. The Markovian framework provides a well-defined and tractable set of design parameters, and a well-defined measure of the performance for any given choice of the design parameters, therefore controllability is empirically verifiable. It is adjustable because the derivation of the design parameters are automatic and computationally tractable, once a new high-level mission objective is given, or the telescope-environment system changes. Finally it is 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 depends only on its current state and is independent of its earlier history. This property is 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  $\mathbb{P}(X_{i+1} = y|X_i = x, a)$ , the conditional probability of transition from state  $x$  to state  $y$  given action  $a \in \mathcal{A}$ . Finally 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*, (i.e. it is possible to be taken at  $t_i$ ), and *progressively measurable*, (i.e. depends only on the current state of the system,  $X_i$ , neither on the future states, nor on any information about the system's history).

To schedule a telescope is to take an admissible action (e.g. to determine the next observation) at all decision steps  $t_i$ . Notice that the decision steps  $t_i$ , are not uniformly spaced, and are determined by the time that each observation takes. **Also note that the scheduling is a finite time procedure from  $t_0$ , until the operation of the telescope is over,  $T$ .** Our framework is general and covers all possible discretizations of the time in both finite and infinite horizon scheduling tasks.

**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 law that for all possible  $x_i \in \mathcal{S}$  suggests an admissible action. The policy is the heart of the scheduler which takes the necessary information encoded in the current state and makes a decision for the next observation. Design of an optimal scheduler is mainly to find an optimal policy.

**Definition 5.** A *deterministic optimal policy*  $\pi$  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.

With this definition, we take the optimal policy to be a policy that maximizes the expected discounted sum of the rewards. The discount factor,  $0 < \gamma \leq 1$ , determines the priority of the overall gain (after an episode of observation) versus instant gains (after a single observation). Larger discount factors prioritize overall gains over instant gains. The choice of  $\gamma$  depends on the application and is empirically tuned.

**Proposition 1.** For the Markov decision process of  $\langle \mathcal{S}, \mathcal{A}, P_a(\cdot, \cdot), R_a(\cdot, \cdot), \gamma \rangle$ , 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 of the following form,

$$\Phi(x_i) = -R_{\pi^*(x_{i-1})}(x_{i-1}, x_i) + \gamma E_{\pi^*}[\Phi(X_{i+1}) | x_i]. \quad (3)$$

*Proof.* See Appendix.

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 scheduler.

As a result of Proposition 1, search for the optimal policy of Problem (1), can be reduced from search over the set of policies (all possible mappings from the state-space to the action-space) to search over  $\Phi$  functions, without loss of generality. This is a significant reduction made possible by the choice of Markovian Decision framework.

## 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. The maximal definition of 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  $N_f^{1000}$ , where  $N_f$  is the number of tessellation centers on the visible sky. For any  $N_f$ ,  $N_f^{1000}$  number of scenarios 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 state of the system with only the most determining information. Thus, the system of telescope-environment is only an approximated Markovian system once its state-space is reduced to a *feature-space*.

On the other hand, Proposition (1) shows that the search for an optimal scheduler lies within the set of functions instead of a much larger set of mappings from the state-space to the action-space. Despite this reduction, Problem (1) is still an infinite dimensional optimization problem, because its variable is a function. To be able to numerically compute the  $\Phi$  function, we assume that it can be expressed as a linear weighted summation of some basis functions. With this structure of  $\Phi$ , our optimization would be further reduced to find the optimal values of the weights.

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

where  $\theta$  is the vector of variables which fully characterizes  $\tilde{\Phi}(.)$ , and  $\Phi_i(x_i)$ 's are the building blocks of  $(\Phi)$ . We refer to  $\Phi_i(x_i)$ 's as *basis functions* which are hand-crafted functions of the features, and are designed in such a way that the domain-knowledge of astronomical observation is incorporated into the decision making strategy.

With this approximation, search-space 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,

$$\tilde{\pi}_\theta^*(x_i) = \arg \min_{a_i} E[\tilde{\Phi}_{\theta^*}(X_{i+1})|a_i] = \arg \min_{a_i} \sum_{j=1}^k \theta_j^* E[\Phi_j(X_{i+1})|a_i], \quad (4)$$

where  $\theta^*$  is a solution to the following optimization problem, in which policy  $\pi$  is fully determined by  $\theta$ .

$$\underset{\theta}{\text{maximize}} \quad E_{\pi_\theta} \left[ \sum_{i=0}^N \gamma^i R_{\pi_\theta(X_{i-1})}(X_{i-1}, X_i) | x_0 \right]. \quad (5)$$

Note that the computational time to find the optimal policy could be relatively long, but it is an offline task and can be done before the telescope starts to operate. However, evaluating the action, given a policy at all  $t_i$  must be at least as fast as the length of the shortest observation. A linear policy not only demands relatively small computational resources to be optimized, but also is very quick to be evaluated in real time.

## 3. SCHEDULER OPTIMIZATION

In this section we introduce two different approaches to solve Problem (5). The solution of this problem is an optimal set of weights,  $\theta^*$ , for a given set of basis functions. The first optimization approach is faster, but requires the high-level mission objectives to belong to a certain class of functions. The second optimizing approach is applicable to all types of the high-level mission objectives, however requires more computational resources.

### 3.1. Reinforcement Learning

Assume that there exists a well-defined notion of an instant reward for each state transition, then  $\Phi(x_i)$  by definition given in Equation (3) is,

$$\Phi(x_i) = -R_{\pi^*(x_{i-1})}(x_{i-1}, x_i) + \gamma E_{\pi^*}[\Phi(X_{i+1})|x_i]. \quad (6)$$

Accordingly, for the parameterized  $\Phi$  function, we require the following,

$$\tilde{\Phi}_{\theta^*}(x_i) = -R_{\pi_{\theta^*}(x_{i-1})}(x_{i-1}, x_i) + \gamma E_{\pi_{\theta^*}}[\tilde{\Phi}_{\theta^*}(X_{i+1})|x_i]. \quad (7)$$

The main idea behind reinforcement learning is to optimize  $\tilde{\Phi}_{\theta^*}(\cdot)$  while the decisions are being made in a simulated environment. First, the policy is initialized by an arbitrary set of variables,  $\theta^0$ . Then the first decision is made by the policy associated with  $\theta^0$ . Then based on the outcome reward, we update the initial set of variables to get  $\theta^1$ . Next, decision is made with a policy associated with the new set of variables  $\theta^1$  which yields a reward. By repeating this process, we gradually update the variables in each decision step until they converge to an optimal value. The optimal  $\tilde{\Phi}$  respects Equation (7) 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 for making the decision. On the other hand,  $\Phi_{\theta^i}(x_i)$  can be approximated by  $E_{\pi_{\theta^i}}[\Phi_{\theta^i}(x_i)]$  which is also evaluated in the decision making process, where  $\theta^i$  is the last version of the optimization variables at  $t_i$ . Using the target value given in Equation (7), the update rule is as follows,

$$\tilde{\Phi}_{\theta^{i+1}}(x_i) = (1 - \alpha)\tilde{\Phi}_{\theta^i}(x_i) + \alpha(\gamma E_{\pi_{\theta^i}}[\tilde{\Phi}_{\theta^i}(X_{i+1})|\pi_{\theta^i}(x_i)] - R_{\pi_{\theta^i}}(x_{i-1}, x_i)), \quad (8)$$

in which,  $0 < \alpha < 1$  is the learning rate. The first term in the right hand side of the equation is the most recent approximated value of  $\tilde{\Phi}$  associated with  $\theta^i$ . This term has  $(1 - \alpha)$  amount of contribution to  $\tilde{\Phi}$  of the next iteration. The second term is the value of the target  $\tilde{\Phi}$  (according to Equation (7) which has  $\alpha$  amount of contribution to  $\tilde{\Phi}$  of the next iteration. Clearly with smaller  $\alpha$ 's this update imposes smaller adjustments from one decision to the next decision. **The choice of  $\alpha$  depends on the application. Higher  $\alpha$ 's are computationally preferred because they speed up the optimization, however depending on the natural dynamic of the system and how much it changes from one decision to another, higher  $\alpha$ 's could make the process of optimization unstable, and diverging.** Accordingly, updates of the variables  $\theta_j$ ,  $j = 1, \dots, k$  can be expressed as follows,

$$\begin{aligned} \theta_j^{i+1} &= \theta_j^i + (\tilde{\Phi}_{\theta^{i+1}}(x_i) - \tilde{\Phi}_{\theta^i}(x_i))\Phi_j(x_i) \\ &= \theta_j^i + \alpha(-\Phi_{\theta^i}(x_i) + \gamma E_{\pi_{\theta^i}}[\tilde{\Phi}_{\theta^i}(X_{i+1})|x_i] - R_{\pi}(x_{i-1}, x_i))\Phi_j(x_i). \end{aligned} \quad (9)$$

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

Recall that in order to be able to use the TD reinforcement learning method, it is necessary to have a well-behaved notion of a reward that reflects the instant gain of all possible decisions at all of the decision steps. Moreover, the discounted sum of the instant rewards has to reflect the objective of the mission according to Equation (1). 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 operated. This however does not reflect all aspects of the high-level mission's objective such as the need to re-observe a field within a valid time window (explained in Section 4), for which there is no equivalent instant reward. For this reason, we also introduced a black-box function optimizer in

the following section. We use the black-box approach when there is no well-defined notion of an instant reward with which we can build the high-level mission’s objective function.

### 3.2. Global Optimization

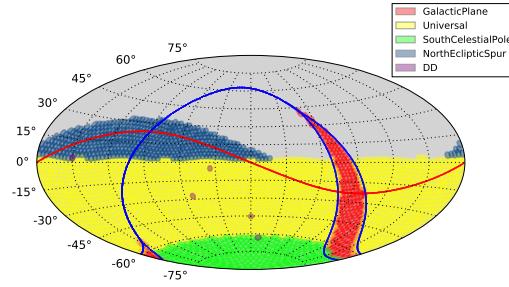
In the absence of a well-defined instant reward, instead of solving problem (5), The following problem can be solved,

$$\underset{\theta}{\text{maximize}} \quad U_{\pi_{\theta}}(x_i, x_{i+1}, \dots, x_j), \quad (10)$$

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  $t_j$ . Unlike the previous method, the policy,  $\pi_{\theta}$ , is fixed during an episode of simulation in this approach. The simulation episode is usually a small fraction of the total time of the telescope’s operation,  $T - t_0$ . The idea is to test the performance of a set of candidate policies in a parallel manner, within a short episode. Then infer a better set of candidates and repeat until the performance cannot be improved. In general,  $U(\cdot)$ , can not be explicitly expressed as a function of  $\theta$ , 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 THE 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 goals and constraints, defined and precisely described in (Ivezic et al. 2008). The notion of the features, enables the scheduler to system-



**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).

atically fetch all of the various information and turn them into comparable quantities for the purpose of the decision making. 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

**Table 1.** Key terms and notations used in the definition of the features and the basis functions

fields	fixed point configuration on the sky such that the visible sky could be completely covered by pointing the telescope toward all of those directions,
$N_f$	total number of the fields,
$t$	Coordinated Universal Time (UTC),
$\tau_s(t)$ ( $\tau_e(t)$ )	beginning (end) of the night that $t$ lies within,
$\tau_{rise(set)}(i, f, t)$	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 $j$ ,
$\Delta t_f$	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,

sky tessellation. There are 6 choices of filter,  $[u, g, r, i, z, y]$ , for each visit. And finally the time is discretized by the natural timing of the process. In other words, the time interval between two decision steps is the time that it takes to execute the observation in between the two decisions. Given that a consecutive visit of the same field-filter is not allowed in the main survey, 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  $\{(f_1(i, f, t_j) \dots f_7(i, f, t_j)) : i = 1 \dots n_f, f \in \{u, g, r, i, z, y\}, j = 0, \dots N\}$ . With this feature-space the implication of the policy, stated in Equation (4), is as follows,

$$\tilde{\pi}^*(x_j) = \arg \min_{(i, f) \in \mathcal{A}_j} \sum_{k=1}^5 \theta_k^* E_\pi[\Phi_k(X_{j+1})|x_j], \quad (11)$$

where  $x_j = [f_1, \dots, f_7(id(t_j), ft(t_j), t_j)]$  is a 7-dimensional state at  $t_j$ , and  $(i, f)$  is an admissible pair of field-filter. Section 4.3 introduces the constraints under which a field-filter pair is admissible at  $t_j$ . Accordingly,  $\mathcal{A}_j$  is the set of all field-filter pairs that are admissible at  $t_j$ .

In the implementation of the scheduling software we took a modular approach. The expected values of the basis functions,  $E_\pi[\Phi_k(x_{j+1})|x_j]$  for  $k = 1, \dots, 5$ , are evaluated in separate software routines. Then they are delivered to the scheduler at the stage of the decision. The basis functions that address the environmental parameters are developed by the LSST community. For example, see (Gressler et al. 2014) for the parameters that capture the status of the LSST site, (Sebag et al. 2008) and (Sebag et al. 2007) for cloud cover measurements that were used to develop a predictive cloud model, and see (Yoachim et al. 2016) for the sky brightness model.

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 observational quality of a field-filter at the time of decision, such as the expected depth of the resulting image.**
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. For the full definition of the basis functions of the Feature-Based scheduler for LSST refer to Section 4.2.

In what follows, first we precisely define the features and basis functions of the LSST scheduler, then we show how the two optimization procedures, described in Section 3, are applied. The optimization results are associated with two sample objective functions. However, the LSST community, and principally any individual can design their own mission objective function, and repeat the optimization procedure to obtain a scheduler that is optimizes the probabilistic expectation of their objective function.

#### 4.1. Features of the telescope scheduler

For designing the features, it is important to avoid redundancy in the information which features contain. It is also critical to hold a modular approach in the delivery of the information to the decision stage. For instance, consider the amount of time,  $\Delta t$ , it takes for a telescope to move from a visit to another. In the LSST problem,  $\Delta t$  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. Conceptually, modular approach makes it possible to track the effect of the operational costs of a similar nature in the overall performance of the scheduling. Particularly, the operational cost is independent of the amount to which each cause contributes to the overall  $\Delta t$ . Hence, bringing the timing of each procedure separately in the decision making level adds unnecessary complications to the design.

This section proposes seven features for the description of the LSST-environment state in Table 2, with the key terms and notations defined in Table 1. Features are designed to efficiently carry the determining information with a modular approach. Each feature is denoted by  $f_k(i, f, t)$  for  $k = 1..7$ , and indexed by the triplet of  $(i, f, t)$ , field, filter, and time. To make a decision at  $t$ , the scheduler computes all of the seven features for all of the  $(i, f)$  pairs. 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 routines to avoid redundant computations. **For features such as the cloud coverage that are inherently random variables, we have separate predictive modules that evaluate the expectation of their values at the time of the next observation.**

The main computational time during the real-time scheduling is spent on the evaluation of the features (evaluating the policy, is only a number of multiplications and summations). Some features such as the slew times are essentially look up tables. Some features, such as the co-added depth are stored in the memory, indexed by a pair of field-filter, which

**Table 2.** 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
$f_1(i, f, t)$	$(slew(id(t), i) + settling(id(t), i) + \Delta t_f I_{f \neq f} \vee t_{dome}(i))$ : 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 limit, 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 $t$ .
$f_6(i, f, t)$	$5\sigma$ -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 $t$ .

the scheduler updates for only one field-filter after every decision step. And the rest are continuous variables such as the location of the clouds which can be interpolated from their past values, even if the exact value at a given time is not available.

#### 4.2. Basis functions of the telescope scheduler

Basis functions,  $\Phi_k$  for  $k = 1 \dots 5$ , are fully determined by the value of the features. Hence, they are indexed in the same way that features are, by a triplet of  $(i, f, t)$ , field, filter, and time. Similar to the update of the features, all five basis functions should be evaluated for all pairs of  $(i, f)$  for a decision at time  $t$ . Except that in this case, admissibility of a field-filter can be evaluated because it only depends on the values of its features which are already computed at this stage (see Section 4.3 for the list of constraints which ensure admissibility). While it is required to evaluate the features for all possible pairs of  $(i, f)$  at all decision steps, the number of the basis function evaluations is on average a factor of three times less than the number of the feature computations.

*Common basis functions* are shared amongst all of the regions of the sky. They are designed to reflect the five general decision factors described in Section 4. The exact definitions of the common basis functions are presented in Table 3, and the key terms and notations can be found in Table 1.

The LSST's mission poses different requirements on different regions of the sky, shown in Figure 1. First we modify  $\Phi_2$ , 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<sup>2</sup> 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 at the same night. There are two regimes to prioritize these fields, encoded in  $\mathbb{1}_{(f_4 \geq W_2)}$ . This function is zero if the expected remaining time for a field to become invisible,  $f_4$ , is less than  $W_2$ , hence the associated cost is minimal ( $=0$ ) to ensure that this field will receive its second visit before it becomes invisible. If  $f_4$  is larger than or equal  $W_2$  it means that the expected remaining time for the field to become invisible is longer than the revisit deadline, therefore  $\mathbb{1}_{(f_4 \geq W_2)}$  is one and the cost is evaluated by a different regime defined by  $\Phi^{\text{pair}}$  in the following.

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

<sup>2</sup> Later, during the development of the scheduler software it was decided that this constraint should be applied to all of the regions.

**Table 3.** 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 $(i, f)$ . scale factor, $s_1$ , is chosen to be 0.43. This is based on the values of $f_1$ in our simulations, and is chosen to ensure that 80% of the $\Phi_1$ values are between 0 and 1. Scaling is a regulation that improves the rate of training convergence
$\Phi_2(f_2(i, f, t))$	$\begin{cases} 0.5, & \text{if } \sum_f f_2(i, f, t) = 0 \\ 1, & \text{if } \sum_f f_2(i, f, t) = 2, \\ 0, & \text{else,} \end{cases}$ reflects the short term visit/revisit priority of field $i$ , conditioned on the total number of the previous same-night visits.
$\Phi_3(f_5(i, t))$	$(1 - \frac{f_5(i, f, t)}{\max_i \max_\phi f_4(\ell, \phi, t)})$ , 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 $5\sigma$ -depth of all $(i, f)$ pairs at $t$ . $\Phi_4$ 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.

where,  $\Phi^{\text{pair}}(f_3) = \exp(-\frac{\min_\phi f_3(i, \phi, t)}{W_2})$ . Fields that have received their first visit of the night are distinguished via  $\sum_f f_2 = 1$ , sum of the number of visits in all filters, then are prioritize amongst themselves by  $\Phi^{\text{pair}}$ . This function assigns a smaller cost to those fields that their first visit occurred earlier than the others.

Deep Drilling Field (DDF) region contains a very small fraction of the visible sky's area (it is about 10 individual fields). Hence it is unnecessary to adjust the basis functions that yield a separate generic policy for such fields. Instead, we treat the observation of DDFs as interruptions to the scheduler's regular operation, with each interruption comprising a sequence of DDF observations. This scheme is computationally more efficient and reduces the structural complexity of the scheduler. The recoverability attribute of the Feature-Based scheduler enables the scheme of the interruptions to be a part of the optimal scheduling.

#### 4.2.1. Controllability of the scheduler

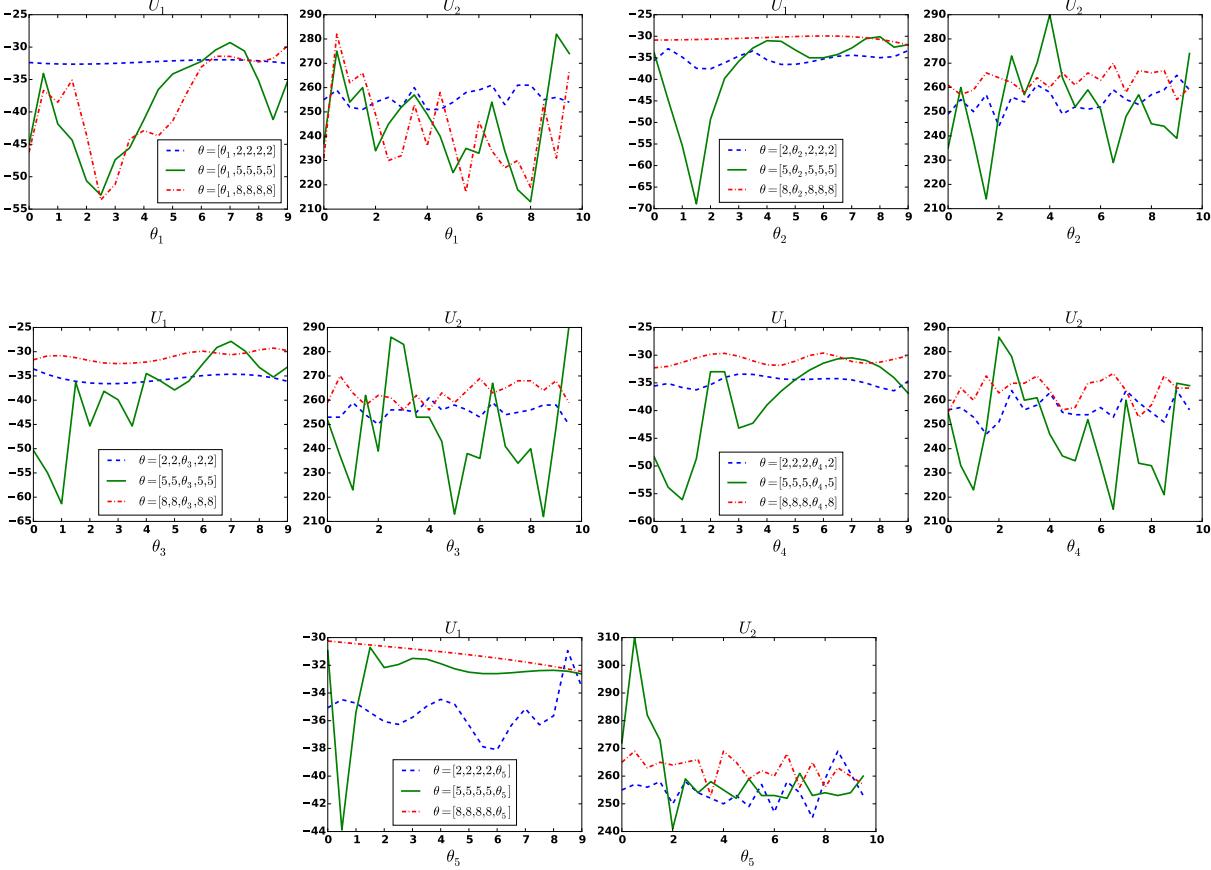
As discussed in Section 2, the scheduler is optimizable if the telescope is controllable given a structure for the scheduler. In this section we discuss the empirical controllability of the LSST given the Feature-based scheduler. What we observe is the variations of two sample objective functions with respect to the variations of the design parameters,  $\theta$ . If there is no meaningful variations then the telescope is not controllable with this scheduler. As a result, the scheduler is not amenable to any form of optimization. On the other hand, if the objective function is extremely variable with respect to the changes in the design parameters, the solution of the optimization is not fully reliable, because the objective is not a well-behaved function of the optimization variables.

In order to observe the variability of the objective functions with respect to changes in the scheduler's design parameters, first we defined a sequence of equidistant values for  $\theta_i \in \{0, 0.5, \dots, 8.5, 9\}$  and kept the other  $\theta$ 's fixed at 2 or 5 or 8 in separate experiments. Then we scheduled a simulated episode of observation for  $t_n - t_0 = 4.8$  hours, for each  $\theta_i$  separately. Finally, we evaluated the objective functions  $U_1$  and  $U_2$  of each schedule. Where,

$$U_1(x_0, x_1, \dots, x_n) = \sum_{\{i: t_0 < t^i < t_n\}} -\text{slew}(id(t_i), id(t_i)) - 10\text{am}(id(t_i)), \quad (12)$$

$$U_2(x_0, x_1, \dots, x_n) = n. \quad (13)$$

The first objective function reflects the slew time, *slew()*, and airmass, *am()*, aversion, and  $U_2$  reflects the time-efficiency of the operation by counting the total number of observations. Note that the second objective function belongs to the class of black-box functions, and the first one belongs to the other class where there exist a decomposition based on the discounted sum of the rewards.



**Figure 2.** One dimensional slices of two objective functions,  $U_1$  and  $U_2$ , defined in Equations (12) and (13) respectively. The variation of the objective functions, specially in the mid-range slices (solid line) suggests that the LSST is controllable with the Feature-Based Scheduler.

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  $\theta$ , which can be an evidence of the controllability. Moreover, the smaller variations for slices closer to the 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 ( $i, f$ ). Feasibility of a candidate  $(i, f)$  is driven by the following factors:

**Table 4.** Feasibility of field-filter  $(i, f)$  for a visit at  $t^{n+1}$ . We use the probabilistic expectation,  $E[\cdot]$ , of the stochastic values, 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 (e.g. by the moon) 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 except DDF
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 <sup>a</sup>
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

<sup>a</sup>Later, during the development of the scheduler software it was decided that this constraint as well as the 7th constraint should be applied to all of the regions.

- Visibility: The candidate field-filter has to be visible.
- Quality: The expected observational quality of a field-filter, such as the expected depth of the resulting image, has to be better than a given threshold.
- Survey’s timing: The science-driven revisit constraints has to be respected.

To ensure measurability, the above criteria must be evaluated based only on the information that is encoded in the feature-space. Exact expressions of the proposed constraints for the LSST scheduler are presented in Table 4.

#### 4.4. Scheduler optimization

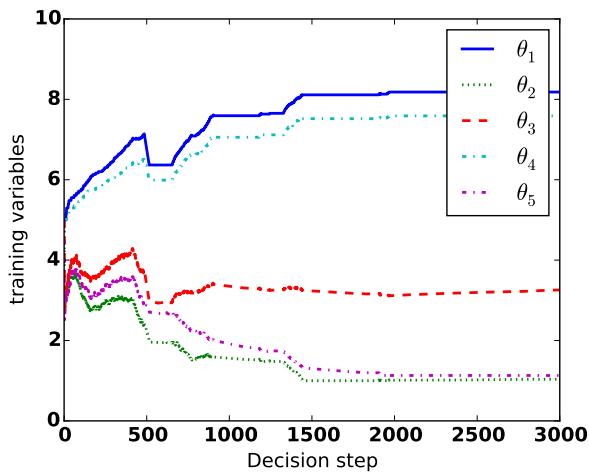
In this Section, we present two simple choices for the high-level mission objective to demonstrate the application of the proposed optimization approaches, discussed in Section 3. (More sophisticated mission objective functions can be defined based on the LSST performance studies such as (Grav et al. 2016), (Graham et al. 2018), (Jacklin et al. 2017), and (Oluseyi et al. 2012)). The choice of optimization algorithm depends on the nature of the mission objective. The first experimental mission objective function in this section can be expressed as the discounted sum of instant rewards  $R(s_{i-1}, s_i)$ , thus the reinforcement learning is applied to find the scheduler’s parameters  $\theta$ . The second objective function cannot be decomposed as a discounted sum of instant rewards, thus we used the global optimizer approach. From the computational point of view, the first approach is preferred. For the following experiment, the reinforcement learning is about 10 times faster than the global optimization, and requires 50 times less memory. From the practical point of view however, for some missions, it is impossible to define an objective function that can be expressed as discounted sum of a well-defined instant reward. In which case, the mission objective can be optimized only via global optimizer.

In the following experiments, for both of the optimizations we used a simulated model of the telescope (Connolly et al. 2014) 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.

#### 4.4.1. Reinforcement learning for the first choice of mission objective.

Let the instant reward,  $R(i-1, i)$ , be  $-\text{slew}(id(t_{i-1}), id(t_i)) - \text{am}(id(t_i))$ . It is defined as a linear combination of the slew time to point the telescope from the  $(i-1)$ th field to the  $i$ th field, and the airmass of the destination. Since both factors have negative effect on the quality of the observation, the reward would be measured by the negative of each. Then the mission objective function can be simply defined as  $\sum_{i=0}^N \gamma^i R(i-1, i)$ .

The simulation for the reinforcement learning starts at  $t_0 = 2462867.5$  *mjd* (2021 January 1), with  $\theta^0 = (5, 5, 5, 5, 5)$ , initialized at the mid-range values, and continues until  $\theta$  converges. Figure 3, is the training curve for all of the variables over a course of 3000 decisions, . 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, after each decision step there is a reward associated with the outcome of the decision which implies a small adjustment on each of the five design parameters. Then the next decision will be taken with a slightly different set of parameters. This procedure continues until the adjustments on the variables are negligible.

For the above choices of reward, learning rate, discount factor and initialization,  $\theta$  converges to  $\theta^* = (8.18, 1.04, 3.26, 7.59, 1.13)$ . With a personal computer<sup>3</sup>, each decision and its associated update takes about 0.8 sec<sup>4</sup>, thus the time of the convergence for the simulation, presented in this section is  $3000 \times 0.8$  sec = 40 minutes . Note that the optimization time is linear with respect to the number of decision steps.

#### 4.4.2. global optimization for the second choice of mission objective.

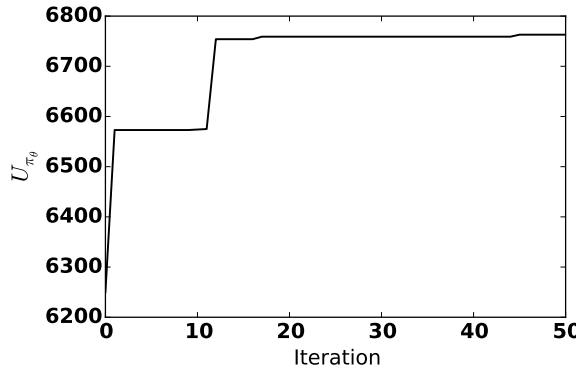
<sup>3</sup> Processor: 1.6 GHz, Memory 1600 MHz DDR3

<sup>4</sup> This reflects the efficiency of the first scheduler’s prototype. Time of each decision simulation for the current efficient software is about 0.036 sec.

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 a given  $t_i$  to a given  $t_j$  which can be expressed as,  $U_{\pi_\theta}(x_i, x_{i+1}, \dots, x_j) = j - i$ . To find a set of parameters,  $\theta$ , that optimizes this objective function we applied the global optimization approach, explained in Section 3. To decrease the computational time, we used the following regulatory constraints,

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

We defined the above objective function,  $U_{\pi_\theta}$ , over a period of 10 days, from  $t_i = 2462867.5 \text{ mjd}$  (2021 January 1) to  $t_j = 2462877.5 \text{ mjd}$  (2021 January 11). Figure 4, shows the value of this objective function at each iteration of the eDE algorithm. The solution  $\theta^* = (1.00, 0.84, 0.99, 1.34, 3.04)$ , yields the best  $U_{\pi_\theta}$  after 50 iterations.



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

eDE is a population-based metaheuristic algorithm. For our experiment, shown in Figure 4, the number of population  $N_P$  is set to be 50. Each function evaluation is in fact, the simulated operation of the telescope for 10 nights, with a candidate scheduler which takes about 8 minutes. Therefore each iteration of the algorithm eDE, takes  $N_P * 8$  minutes<sup>5</sup>. 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

<sup>5</sup> Processor: 1.6 GHz, Memory 1600 MHz DDR3

In this section, the LSST Metric Analysis Framework (Jones et al. 2014) is used, to compare<sup>6</sup> the performance of a modified version of the Feature-Based scheduler with opsim V4 and opsim V3, the most recent producers of the baseline schedules of the LSST. (We use two specific sequences produced by each of these scheduler. Namely, astro-lsst-01-2013 and minion-1016 produced respectively by opsim V4 and opsim V3.)

**The following results are based on the simulations that use a computer model of the LSST-environment. First we optimized the scheduler with the model over a limited simulated episode of observation. Then we used the optimal scheduler to schedule the same model of the LSST-environment within a simulated period of time from 2021 to 2031 for the purpose of performance evaluation.** The resulting schedule however, will not be used in practice, because it is made by some realization of the random processes such as the cloud coverage. For the real-life operation of LSST, we use the same approach to optimize the scheduler, however the decisions are made on-the-fly, using the real-time values of the features.

Recall that the optimization part consumes the main portion of the total computational time. Fortunately, the scheduler’s optimization is an offline process, and can be done before the telescope is ready to operate. On the other hand, the real-time decision making, with an already optimal scheduler is a very fast process. It only includes evaluation of the features and basis functions, in addition to basic summation and sorting operations. The current version of the Feature-Based scheduler can schedule 28 observations per second, while the length of each observation in real-time is at least 30 seconds. Hence, the decision making speed is roughly 900 times faster than what is required for the real-time scheduling. However, we are still working to increase the time efficiency of the software, because it directly effects the computational time of the offline optimization procedure.

The *Modified Feature-Based* scheduler is under active development<sup>7</sup>, 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.

Unlike the default sky tessellation for LSST, we do not require the tessellation centers to be determined based on the telescope’s field of view. The default sky tessellation adopted in the baseline scheduler results in 23% of the sky being covered by more than one field which causes serious non-uniformity in the final coverage of the sky. In the modified Feature-Based scheduler, we adopt a 50 times finer discretization of the sky using Hierarchical Equal Area isoLatitude Pixelization (HEALpix) (Górski et al. 2005). The fact that the decision-making is not computationally expensive makes it possible to use a much finer discretization of the feature-space resulted by the finer tessellation of the sky. This approach allows the scheduler to handle the overlaps which cause inhomogeneity of the final sky coverage.

In addition to adopting a finer discretization, we use a spatial dithering scheme to randomize the final pointing of the telescope by a small amount around the tessellation centers to further assist the homogeneity of the coverage. 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). Moreover, the modified Feature-Based scheduler uses separate routines to decide if an observation will need to be observed in a pair, and to decide if a Deep Drilling sequence should be executed by interrupting the normal operation of the telescope.

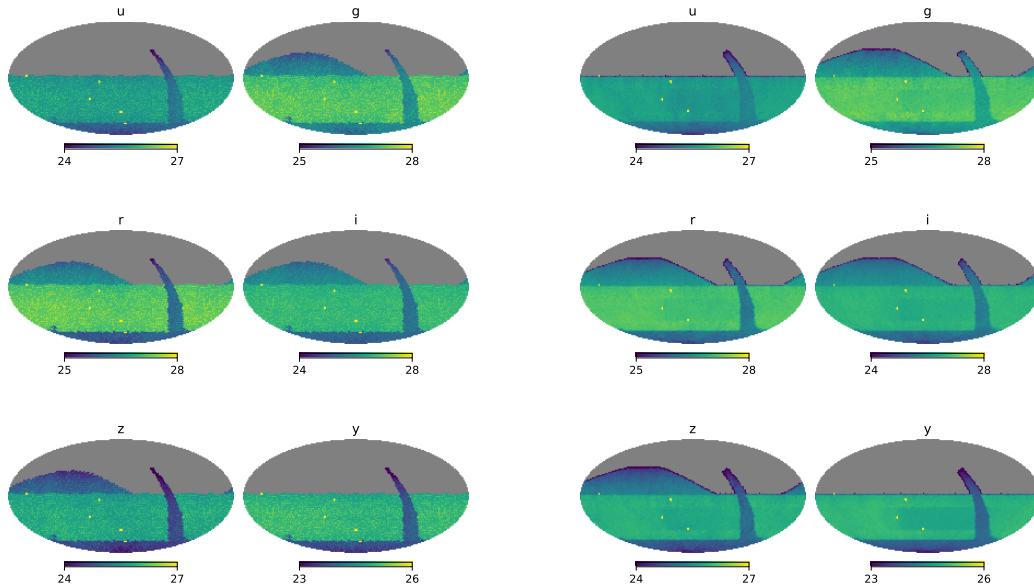
<sup>6</sup> The sky background models and weather downtime used to benchmark the algorithms are not exactly identical because of the practical difficulties in the separation of the environment and the opsim scheduler. However, for the purpose of the comparisons in this section, the behavior of our sky and observatory model is sufficiently close to the official model. See (Delgado & Reuter 2016), and (Reuter et al. 2016) for the official operation simulator.

<sup>7</sup> GitHub repository: [https://github.com/lsst/sims\\_featureScheduler](https://github.com/lsst/sims_featureScheduler).

### 5.1. Sky coverage uniformity

For a survey telescope, such as the LSST, the density of the co-added depth over the visible sky should ideally be 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 full sky map. Figure 6, demonstrates the smoothness of the coverage by zoom-in to a smaller area of the sky map. It shows the co-added depth values of opsim V4, with and without dithering, and compare them with that of the modified Feature-Based scheduler in the  $r$  band. The zoomed-in area is around the boundary of WFD and GP regions which have different target co-added depth. 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 the dithering which takes place after the decision is made.

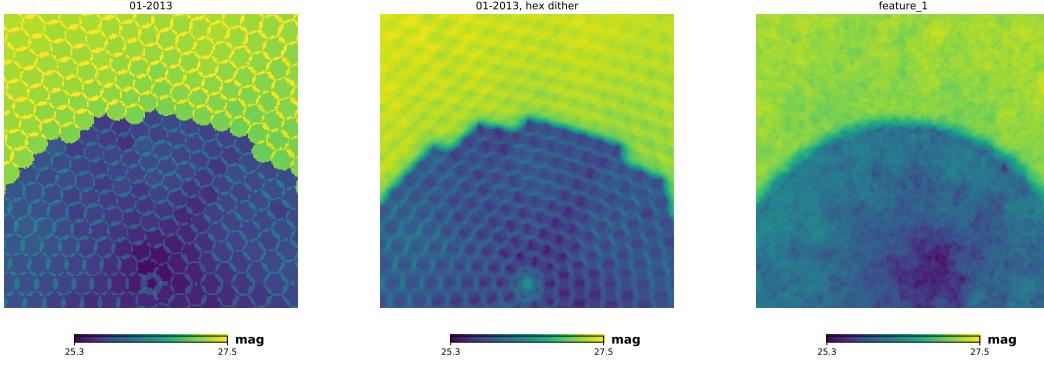
Figure 7 compare the distribution of the co-added depth on a (finely) discretized sky. The modified Feature-Based scheduler has paved the left-most peak that appears in the distribution of opsim V4. This peak is the result of the field's undesirable overlaps. Table 5 contains the median and variance of the co-added depth for both of the schedules in each of the sky regions and in each filter. The modified Feature-Based scheduler provides deeper (higher median), and more uniform (lower variance) coverage in most of the cases.



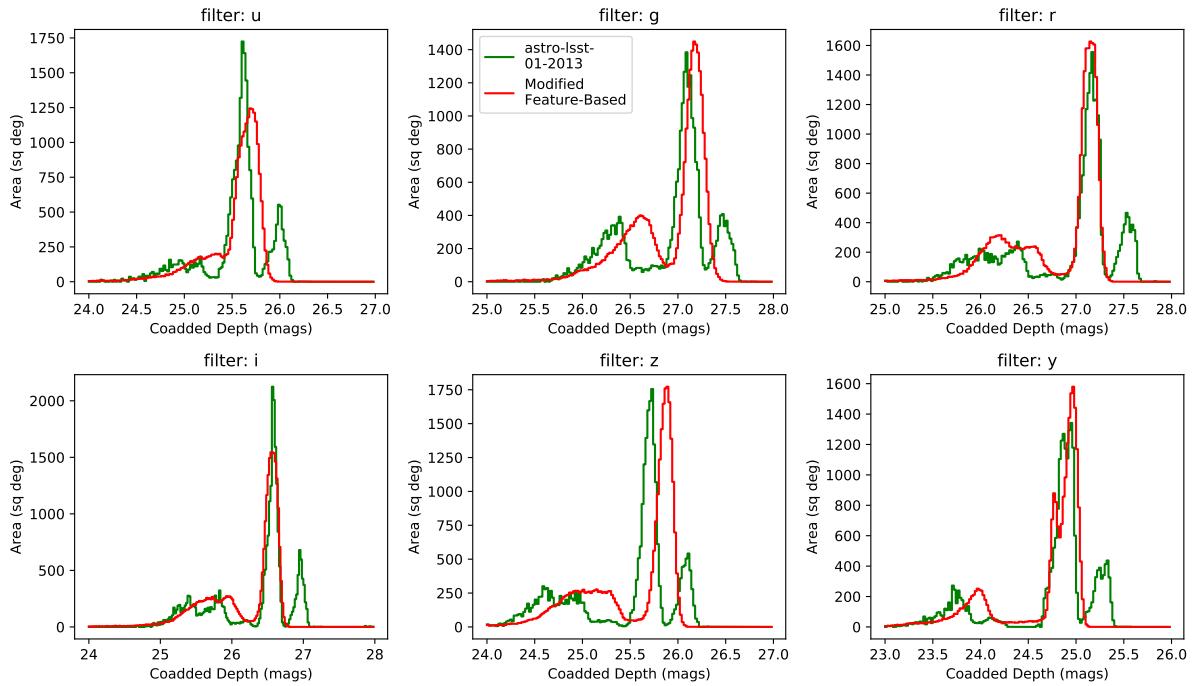
**Figure 5.** Sky coverage in each of the six filters  $\{u, g, r, i, z, y\}$ , measured by co-added depth. According to the mission's objective, the scheduler has to provide a uniform coverage of the visible sky within each region, and in each filter. The left panels show the opsim V4 simulation results while the modified Feature-Based scheduler is on the right. Even without a given observation proposal, modified Feature-Based scheduler can closely match the large-scale footprint of the official survey. (The individual yellow dots with high co-added depth are fields in deep drilling survey, and the NES region is not visited in the  $u$ , and  $y$  bands.)

### 5.2. Pairs

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. Because, the moving objects usually belong to the solar system, the pair constraint was initially imposed only



**Figure 6.** Co-added depth coverage in a zoomed-in area of the sky map, around the border between the WFD (green) and SCP (blue) regions, in the  $r$  band. The smooth coverage of the modified Feature-Based scheduler (right) versus the granular pattern of opsim V4 (left), further respects the uniformity of the coverage which is one of the most fundamental objectives of the LSST mission. The middle figure is the coverage of opsim V4, with dithering of the same sequence of observations which fundamentally can not become as smooth as the right figure, because unlike Feature-Based scheduler, the scheme of opsim V4 does not easily allow for decision making with arbitrarily fine tessellations 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 in addition to a finer tessellation of the sky smoothens the density of the coverage where the fields overlap.

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. In addition, identification of the other varying

**Table 5.** 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 opsim V4 in terms of the uniformity of the coverage, with lower variances, specially in WFD and SCP regions.

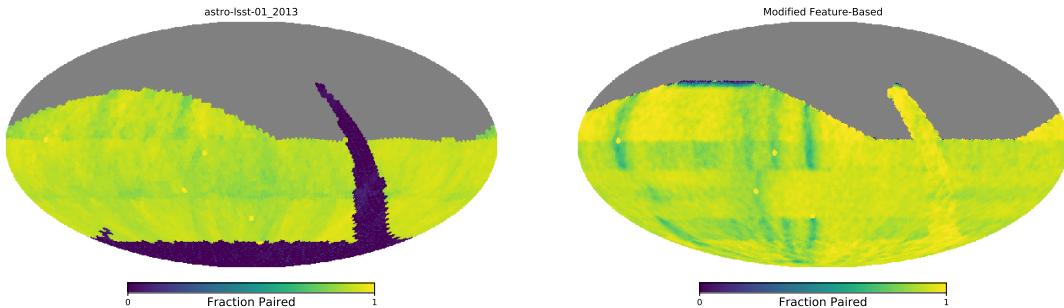
filter	Median, Variance							
	opsim V4				Modified Feature-Based			
	WFD	GP	SCP	NES	WFD	GP	SCP	NES
<i>u</i>	25.63, 0.04	25.12, 0.11	24.91, 0.12	-	25.68, 0.01	25.32, 0.05	25.10, 0.04	-
<i>g</i>	27.13, 0.04	26.41, 0.09	26.32, 0.12	26.30, 0.13	27.18, 0.01	26.69, 0.04	26.56, 0.04	26.47, 0.09
<i>r</i>	27.19, 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, 0.09	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	-

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.

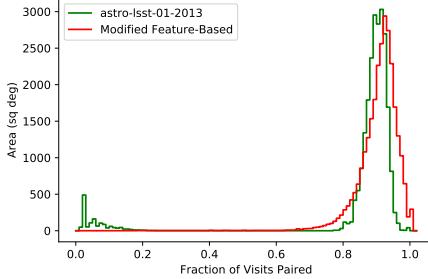
Note that the structure of the Feature-Based scheduler, allows for extension or restriction of the constraints down to the level of an individual field-filter, with neither contradicting any of the Markovian assumptions, nor breaking the structure of the implementation.

Figure 8 demonstrates the distribution of observations in pairs (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 schedulers in satisfying the pair constraint. This success rate ranges from 0 to 1, and the higher values indicate the more successful pair visits.

Figure 9, compares the distribution of the pairs ratio of the modified Feature-Based scheduler and opsim V4 on a (finely) discretized sky. Note that the peak of the distribution for the modified Feature-Based scheduler is closer to 1, which means a larger area of the sky is covered by successful pairs visit, however opsim V4 offers a sharper concentration of the values that can be interpreted as a more homogenous pairs visit which agrees more with the LSST’s mission to survey the sky in a uniform manner.



**Figure 8.** The ratio of the 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 opsim V4 (left), it is applied to the WFD and NES regions only.



**Figure 9.** Distribution of the pairs ratio to the total number of the observations, in the  $g$ ,  $r$ , and  $i$  filters. The Feature-Based scheduler covers a large area of the sky by successful pairs visit (higher median). And opsim V4, maintains a uniform ratio of pairs for a larger area of the sky (lower variance).

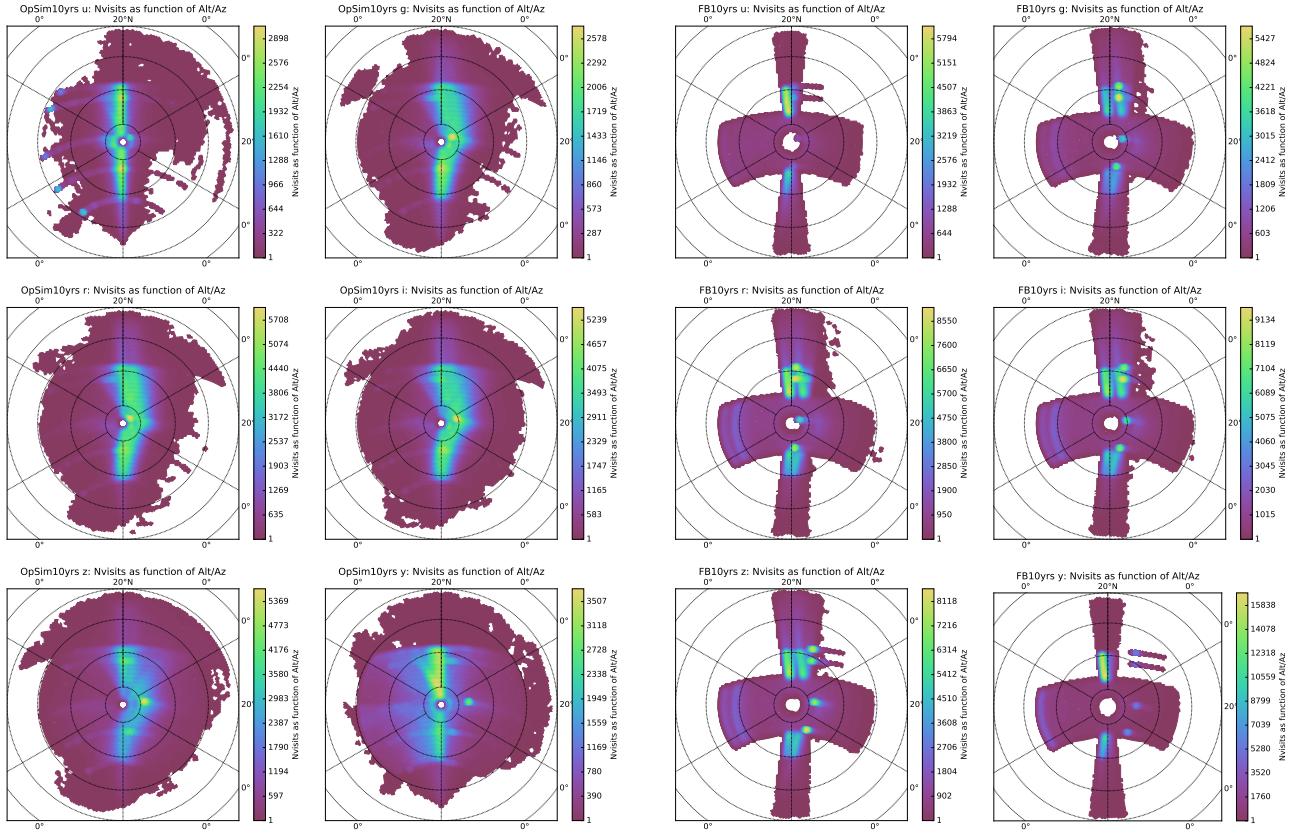
### 5.3. AltAz and airmass distributions

Airmass is one of the major obstacles for ground-based instruments. Zenith observations have the minimum airmass, however off-zenith observations cannot be avoided, due to time-efficiency and design limitations. In which case, observations around the meridian provide high-quality images and consequently result in more efficient operation of the instrument. 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 Feature-Based scheduler schedules more visits around the meridian zone. In addition, it offers a consistent concentration peak on the east wings, which is essential for a higher success rate of the pairs constraint. Because, if the first visit of the night occurs when the field is on the east side of the sky, the scheduler has a longer opportunity to schedule the second visit of the night, hence it is more likely to perform a successful pair visit. Figure 11 demonstrates the density of visits collectively in all filters for opsim V3, opsim V4, and the modified Feature-Based scheduler. Note that the adjustability of the Feature-Based scheduler allows for a significant change in the behavior of the telescope, in this case, to encourage observations around the meridian we defined a new basis function in such a way that the modified Feature-Based scheduler prefers to observe a contiguous set of fields that is then re-observed later in the same order.

### 5.4. Signal-to-Noise ratio

For a multi-objective survey telescope, such as LSST, comparing the overall performance of the different schedules is a difficult task. Particularly because of the large number of competing factors that are involved in the evaluation of their performance. In some cases these criteria are not even objective or well-defined, such as the importance of an area of astronomy compared to the rest. Nevertheless we conclude this section with a general comparison of the performances using the overall signal-to-noise ratio of the schedules. Table 6 reflects the value of median throughput for three different schedulers in  $r$  and  $g$  bands. The modified Feature-Based scheduler significantly outperforms the other two.

In addition note that the throughput is mainly determined by the combination of open shutter fraction (OSF), and airmass. The open shutter fraction is the total time that the telescope camera shutter was open divided by the maximal 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. Comparing the values of the OSF and airmass for both of the baseline schedules, opsim V3 and opsim V4, shows that there is a trade-off between the two values. While opsim V4 offers better median airmass, its OSF is worse. However, its median throughput is very close to that of opsim V3. This comparison reveals that



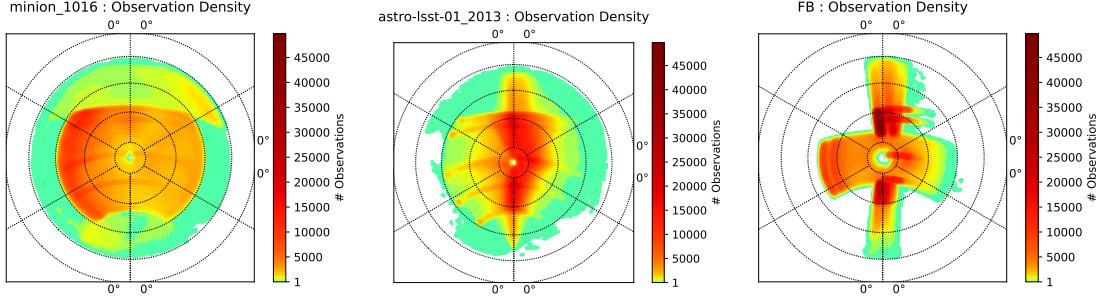
**Figure 10.** Each plot is the distribution of the visits on an altitude-azimuth sky map in one of the six filters. The two left columns belong to opsim V4, and the two right columns belong to the modified Feature-Based scheduler. The higher concentration on the meridian (vertical axis) for the modified Feature-Based scheduler shows a more desirable behavior. Moreover, consistent concentration of the visits on the east wing can potentially provide a better success rate in pairs observation.

**Table 6.** Comparison of the schedulers in a section of the LSST main survey area, WFD

Survey	median throughput <i>r</i> (%)	OSF <i>g</i> (%)	median Airmass	dithered
modified Feature-Based	63.7	47.0	0.705	1.1
opsim V3	55.3	40.0	0.736	1.2
opsim V4	54.4	40.8	0.715	1.1

the change of meta-parameters, and tweaking the structures of proposal-based schedulers such as opsim V3 and opsim V4, only changes the balance of trade-off between OSF and airmass, but not the actual performance of the scheduling.

## 6. CONCLUDING REMARKS



**Figure 11.** The distribution of the visits on an altitude-azimuth sky map for three schedulers. On the left, opsim V3 schedules most observations at high airmass and very few on the meridian. In the middle, opsim V4 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.

This study demonstrated that a Markovian scheduler with expert-designed features, and a parametrized linear 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, while proposals are designed for feasibility, our approach is designed for optimality in addition to feasibility. In general, automatic schedulers such as the Feature-Based scheduler, are fundamentally less prone to suboptimality compare to the schedulers that heavily rely on human interaction. This is mainly due to the manual tailoring which is performed based on the inspections of the instances. Moreover, adjusting the behavior of human-dependent schedulers are inconvenient and time consuming in practice. Furthermore, being modeled as a Markovian Decision Process, the Feature-Based scheduler is a systematic approach to operate the instrument under uncertainties and interruptions.

On the other hand, the building blocks of the Feature-Based scheduler are designed modularly in an intuitive way for the astronomy community. This property allows for expert intervention if needed. However, the robust structure and implementation of the scheduler, allows regulated interventions only. In particular, valid interventions include defining new features, new basis functions, and adjusting the design parameters, and none of which breaks measurability, linearity, and memorylessness of the scheduler. As a result, all of the desirable properties such as 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 provide a user-friendly 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 time efficiency of the different proposed strategies.

Computationally, the optimization part of the scheduler is the most demanding task. Once the optimal values of the design parameters are known, one night of observation can be simulated in 2-3 minutes. Equivalently, in real-time operations, decision are made on-the-fly about 900 faster than what is needed. For the optimization part however, the necessary computational time and resources vary. If many different

objective functions are being tested for planning a mission, then a quick *eDE* optimization, that takes a few hours, 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 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, seasonal variations, and weather patterns, then trains a scheduler specifically for each category to further increase the efficiency of the operation.

## APPENDIX

*Proof.* (of Proposition 1) Consider a function  $C_\pi : \mathcal{S} \rightarrow \mathbb{R}$ , defined as follows,

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

Where  $E_\pi[\cdot]$  is the expectation of the argument inside the brackets, if policy  $\pi$  is being used. Then by applying the law of total expectation on the second term,

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

By assuming a finite state space one can expand the expectation as a finite sum,

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

Then by the Markov property, probabilities that are conditioned on the whole previous sequence of states can be replaced by probabilities that are conditioned only on the previous state,

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

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

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

For the next time-step, one can construct a function,  $\hat{C}$ , 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 (1) 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]. Moreover  $C^*(x_0) = \min C^{\pi}(x_0)$  attains the optimal value by the construction of  $C^{\pi}(x_0)$  which is equal to  $-E[\sum_{\pi(x_i)}(x_i, X_{i+1})|x_0]$ .

Now, given a  $C^*$ , an optimal policy can be simply evaluated,

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

Finally define,

$$\Phi(X_{i+1}) := R_{a_i}(x_i, X_{i+1}) - \gamma C^*(X_{i+1}), \tag{3}$$

and substitute  $R_{a_i}(x_i, X_{i+1}) - \gamma C^*(X_{i+1})$  with  $\Phi(X_{i+1})$  in Equation (2) to complete the proof.  $\square$

**Acknowledgments.** We would like to thank the LSST team, 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 Dr. Michael Reuter for their insightful comments spanning from the idea stage to the end of this project. We are also grateful for the time that the anonymous referees took to provide both high-level and detailed comments which significantly improved the clarity of the paper. This work is financially supported by the National Science Foundation under Cooperative Agreement 1258333 managed by the Association of Universities

for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members. The DIRAC Institute is supported through generous gifts from the Charles and Lisa Simonyi Fund for Arts and Sciences, and the Washington Research Foundation.

#### *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
- Connolly, A. J., Angeli, G. Z., Chandrasekharan, S., et al. 2014, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 9150, Modeling, Systems Engineering, and Project Management for Astronomy VI, ed. G. Z. Angeli & P. Dierickx, 14
- Delgado, F., & Reuter, M. A. 2016, in SPIE Astronomical Telescopes+ Instrumentation, International Society for Optics and Photonics, 991019–991019
- Delgado, F., & Reuter, M. A. 2016, in Proc. SPIE, Vol. 9910, Observatory Operations: Strategies, Processes, and Systems VI, 991013
- 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)
- Graham, M. L., Connolly, A. J., Ivezić, Ž., et al. 2018, AJ, 155, 1, doi: [10.3847/1538-3881/aa99d4](https://doi.org/10.3847/1538-3881/aa99d4)
- Grav, T., Mainzer, A. K., & Spahr, T. 2016, AJ, 151, 172, doi: [10.3847/0004-6256/151/6/172](https://doi.org/10.3847/0004-6256/151/6/172)
- Gressler, W., DeVries, J., Hileman, E., et al. 2014, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 9145, Ground-based and Airborne Telescopes V, ed. L. M. Stepp, R. Gilmozzi, & H. J. Hall, 1
- Gurobi Optimization, I. 2016, Gurobi Optimizer Reference Manual. <http://www.gurobi.com>
- Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, The Astrophysical Journal, 622, 759
- 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>
- Jacklin, S. R., Lund, M. B., Pepper, J., & Stassun, K. G. 2017, AJ, 153, 186, doi: [10.3847/1538-3881/aa64d1](https://doi.org/10.3847/1538-3881/aa64d1)
- 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
- 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
- Oluseyi, H. M., Becker, A. C., Culliton, C., et al. 2012, AJ, 144, 9, doi: [10.1088/0004-6256/144/1/9](https://doi.org/10.1088/0004-6256/144/1/9)
- Reuter, M. A., Cook, K. H., Delgado, F., Petry, C. E., & Ridgway, S. T. 2016, in Proc. SPIE, Vol. 9911, Modeling, Systems Engineering, and Project Management for Astronomy VI, 991125
- 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