The Scheduler development progress and new Feature-based algorithm

Tiago Ribeiro & Lynne Jones
Recently, the LSST Scheduler code was generalized to enable use of different scheduling algorithms.

This generalization was fully implemented on the last release and now we officially support two different algorithms:

- the **Proposal Based Scheduler** (PBS) and
- the **Feature Based Scheduler** (FBS).

The PBS was used to produce the current baseline 2018a simulation and all simulations provided so far for the call for white papers.

The FBS (originally a prototype code) was developed to a point where we can now generate a simulation comparable to the old PBS baseline.

The FBS provides a much more flexible alternative to scheduling than the previous PBS and opens opportunities to explore a new realm of cadences that were hard to realize with the PBS.
Overview

- Bottom-up Overview of the Feature Based Scheduler.
- Top-down Overview of the Feature Based Scheduler functionalities used to simulate a candidate baseline.
- Updates on scheduler development.
Bottom-up overview
The Feature-based Scheduler

- Feature: M5Depth
- Feature: N_obs
- Feature: Slewtime
- Feature: Moon distance
- M5 reward map
- Target reward map
- Slewtime reward map
- Moon avoidance map

Reward Map: Filter r
Features

- Features contain information about the state of the system; they can be any type supported by Python, including arrays, dictionaries or classes.
- They are classified in one of two types: survey features and condition features.
  - **Survey features:**
    - Store information about the survey, e.g. total number of observations, number of observations on a specific filter, number of observations across the sky, observed depth, rotation angle distribution, etc...
  - **Condition features:**
    - Store information about the environment, e.g. weather, sky brightness, airmass, observatory state, slew time, etc...
The Feature-based Scheduler

Conditions

Survey

Feature: M5Depth

Feature: N_obs

Feature: Slewtime

Feature: Moon distance

M5 reward map

Target reward map

Slewtime reward map

Moon avoidance map

Reward Map: Filter r

2.64187 6.61768
Basis functions

- Basis functions compute reward values from features.
- Typically, a basis function takes an array of values representing some property across the sky (the features - often healpix maps) and computes the rewards for each position:
  - e.g. the slew time to reach a position in the sky (slew time feature) is converted into a slew 'reward' by adding a weighting function
  - There are also cases where a basis function will use strings or other complex data types to generate a reward:
    - e.g. the current filter and mounted filters features, plus the time since the last filter change feature, turn into a reward to stay in the same filter or switch to a different filter.
The Feature-based Scheduler

- Feature: M5Depth
- Feature: N_obs
- Feature: Slewtime
- Feature: Moon distance

- M5 reward map
- Target reward map
- Slewtime reward map
- Moon avoidance map

Reward Map: Filter r

PST SCI COLLAB - 09/26/2018
Surveys

- Surveys combine a set of basis functions and weights on their reward values to provide a final reward map for the survey.
- Surveys also contain a list of features that it can use for internal logic that is not exactly captured by reward maps.
- A decision function specific to the survey is then applied to the final reward map to select the next observation(s) (and its reward).
- A survey can return a single observation or a list of observations.
- Example of a survey and decision function:
  - Greedy_survey_fields:
    - From the position in the sky with the highest reward, select the closest “survey field” to observe.
    - The survey field is derived from a sky tessellation using the size of the LSST camera FoV.
    - Each night the tessellation is dithered randomly.
The Feature-based Scheduler

Feature: M5Depth
Feature: N_obs
Feature: Slewtime
Feature: Moon distance

M5 reward map
Target reward map
Slewtime reward map
Moon avoidance map

Reward Map: Filter r

PST SCI COLLAB – 09/26/2018
Core Scheduler

- The core scheduler is responsible for selecting the next observation from a set of surveys.
- Each survey returns a final reward (a floating point number).
- The core scheduler selects the survey with the highest reward and builds a queue with the proposed observation(s).
- If a single observation is provided, a new set of rewards is computed after the observation is taken.
- If a list of observations is provided, the core scheduler works through the list, checking at each observation that the request is still valid (given the observing conditions).
  - If observing conditions change such that the queue cannot continue, the core scheduler clears the queue and starts over.
The Feature-based Scheduler

- Core Scheduler
- Survey 1
- ...
- Survey N
- BF 1
- ... BF N
- Feature 1
- ... Feature N

BF = Basis functions
Top Down overview:
Feature based scheduler candidate baseline
Surveys

- For **each filter** we use a `Greedy_survey_fields` that will take the baseline observations on the footprint (WFD+GP+SCP+NES)
- A `Pairs_survey_scripted`, which adds single observations from griz `Greedy_survey_fields` surveys to a queue to make pairs
- A `Deep_drilling_survey` for each of the deep drilling fields.
Greedy_survey_fields

- Basis functions:
  - HourAngle_bonus_basis_function
  - M5_diff_basis_function
  - Target_map_basis_function
  - MeridianStripeBasisFunction
  - Aggressive_Slewtime_basis_function
  - Goal_Strict_filter_basis_function
  - Avoid_Fast_Revists
  - Bulk_cloud_basis_function
  - Moon_avoidance_basis_function
- Extra features:
  - mounted_filters
Pairs_survey_scripted

- Basis functions:
  - MeridianStripeBasisFunction
  - Moon_avoidance_basis_function
  - Bulk_cloud_basis_function
- Extra features
  - Pair_map
  - current_mjd
  - current_filter
  - Altaz
  - current_lmst
  - m5_depth
  - Moon
  - slewtime
Deep_drilling_survey

- Extra features
  - current_filter
  - mounted_filters
  - observatory
  - night
  - N_obs
  - N_obs_self
  - lmst
  - sun_moon_alt
  - moon
  - last_obs_self
  - last_seq_obs
  - mjd
  - night_boundaries
  - bulk_cloud
M5_diff_basis_function

- Condition Features:
  - M5Depth
- Returns:
  - M5Depth – dark_map
**Target_map_basis_function**

- **User input:**
  - target_map
  - norm_factor

- **Survey Features:**
  - N_obs
  - N_obs_count_all

- **Returns:**
  - target_map * N_obs_count_all * norm_factor - N_obs
Goal\_Strict\_filter\_basis\_function

- **User input:**
  - Goal per filter
- **Condition Features:**
  - Current\_filter
  - Mounted\_filter
  - Sun\_moon\_alts
  - Current\_mjd
- **Survey Features:**
  - Last\_observation
  - Last\_filter\_change
  - N\_obs\_all
  - N\_obs
- **Returns:**
  - A float number between 0 and 1 that depends on the time since last filter change, sun and moon altitudes and the number of observations in each filter according to an input goal
The Feature-based Scheduler

- Feature: M5Depth
- Feature: N_obs
- Feature: Slewtime
- Feature: Moon distance
- M5 reward map
- Target reward map
- Slewtime reward map
- Moon avoidance map
Updates on scheduler development
Updates on scheduler development

- V1.3 officially released on 08/31/2018 (http://ls.st/x3s)
  - Focus of this release was preparations for operations and Scheduler API.
    - API is now fully integrated
    - Capable of doing cold start (rebuild scheduler state by replaying history of observations)
    - Improved control sequence (according to system design)
    - Dome crawling
    - Jupyter notebook integration
    - ...
- Feature-based Scheduler candidate baseline simulation
Conclusions
Conclusions

• The Feature based scheduler provides a flexible framework to develop survey observations.
• Information regarding survey properties and overall conditions are stored in features than can be used by a basis function and/or survey to compute reward and take decisions.
• Complex spatial and time dependent behavior can be realized using easily accessible Python code and logic on basis functions and surveys.
• The Feature based scheduler code is now mature enough to provide a baseline candidate that can be directly compared with previous simulations, improved by community input and modified as scientific priorities changes over time.
Support material for users

- Detailed instructions on how to setup and run OpSim docker container: [http://ls.st/624](http://ls.st/624) (Boberg, O.)
- Feature based scheduler paper: [http://ls.st/ysz](http://ls.st/ysz) (Elahesadt, N.)
- Feature based scheduler candidate baseline (MAF metrics): [http://ls.st/ica](http://ls.st/ica)
- FBS Baseline comparison with PBS: [http://ls.st/y17](http://ls.st/y17)

Thank you!