

Evaluating Scheduling Strategies for JWST Momentum Management

Reiko Rager and Mark Giuliano

Space Telescope Science Institute
3700 San Martin Drive, Baltimore, MD 21218
[rager, giuliano}@stsci.edu](mailto:{rager, giuliano}@stsci.edu)

Abstract

The James Webb Space Telescope (JWST) is a large, infrared-optimized space telescope, designed to find the first galaxies that formed after the Big Bang. As the mission is scheduled to launch in 2013, the mission goals and the primary hardware design are fixed. However, details of the hardware and the implied operating limits are still under development. In particular, JWST requires a very large sun shield to protect its scientific instruments. Handling the movement of this massive shield requires managing the momentum buildup in the reaction wheels used to position the telescope. Accounting for momentum management is expected to be a major constraint driving the efficiency of JWST scheduling. We present a case study showing the utility of alternative constraint mechanisms for handling momentum management during scheduling.

Introduction

The James Webb Space Telescope (JWST) is a large, infrared-optimized space telescope, designed to find the first galaxies that formed after the Big Bang. As the mission is scheduled to launch in 2013, the mission goals and the primary hardware design are fixed. However, details of the hardware and the implied operating limits are still under development. In particular, the final design and operating procedures for the JWST sun shield are still under development. JWST requires a sun shield about the size of a tennis court to protect its science instruments from overheating. Solar radiation pressure on the sunshield causes momentum to accumulate in the spacecraft's reaction wheel assemblies. The wheels have a limited capacity to store momentum, and stored momentum must be dumped using spacecraft thrusters. The resulting usage of non-renewable fuel to fire the thrusters makes momentum management a potential limiting factor in the lifetime of the mission. As the momentum accumulated by an observation varies over time, momentum management is expected to be a major constraint driving the efficiency of JWST scheduling. We examine alternative methods for momentum scheduling including preferences, constraining the time of individual observations, and implementing a momentum resource model. The purpose of the study is multi-directional. First, we provide feedback on the design

of the JWST hardware; Is the current excepted hardware limit feasible for the operational goals? What would be the impact of different hardware limits on scheduling? Second, we investigate scheduling strategies that can be incorporated into the real scheduling system to be built for JWST; We need to find a strategy that works well in any JWST hardware configuration and to find factors that influence the selection of the strategy.

The SPIKE system is used in the evaluation. SPIKE (Johnston and Miller, 1994) is a planning and scheduling tool kit that was created for use on the Hubble Space Telescope and is currently being used for multiple orbital and ground based astronomical missions including FUSE (Calvani, 2004), Chandra, Subaru (Sasaki, 2000), and SIRTf (Kramer, 2000).

The remainder of the paper reads as follows. First, we present details of the JWST mission, its operational constraints, and the momentum constraint. Next, we describe the experimental test-bed including the utilization of SPIKE and the observation set being used. Two classes of experiments are described; (1) scheduling studies that utilize alternate mechanisms for handling momentum, and (2) analytic experiments investigating possible strategies for reducing momentum buildup. Finally, we discuss future work.

JWST Mission

JWST will have infrared sensitive detectors and a 6.5-meter primary mirror designed to look through dust clouds to see the earliest formation of stars and planets. The telescope will have a lifetime of 5 to 10 years and will be placed in an Earth-trailing L2 orbit, 1.5 million km from Earth.

JWST will provide time to general observers through a time allocation board. Approved observers will prepare their programs using an automated tool (Krueger, 2004). Programs will be submitted to the JWST Science Operations Center (SOC) and will be scheduled by SOC staff. JWST will have limited on-board scheduling capabilities that allow it to handle observation failures and unknown observation overheads. Schedules uploaded from the ground system will consist of a time ordered list

of observations where each observation has an earliest and latest start. JWST executes the observations in turn moving to the next observation when the current observation finishes or fails. The action taken by the on-board scheduler depends on the relationship between the current time and the observation's planned early start and latest start. If the current time is before the early start, the telescope will be idle until the early start time. If the current time is after the latest start, the observation will be skipped. Otherwise the observation is executed. Implications of the on-board scheduling capabilities will be considered when evaluating the momentum constraint.

JWST Scheduling Constraints

A scheduling system for JWST has to satisfy several types of constraints on tasks. First, a task has to obey all the requirements specified by the user. Second, any astronomical target can be observed by JWST only at certain times of the year determined by the location of JWST, the sun and the target. We call such time intervals *visibility windows*. A task has to be scheduled within its visibility windows. Third, output schedules must satisfy the limit on momentum accumulation. Each of these constraint types is discussed below.

User specified observation constraints

An observation program may contain the following scheduling constraints:

- **o between t_1 and t_2 :** observation o has to begin between time t_1 and time t_2 .
- **o_1 after o_2 by d_1 to d_2 :** observation o_1 has to begin between $\langle \text{start of } o_2 \rangle + d_1$ and $\langle \text{start of } o_2 \rangle + d_2$.
- **group $o_1 \dots o_n$ within d :** the set of observations o_1, \dots, o_n have to begin within duration d .
- **orientation o_1 from o_2 by a_1 to a_2 :** the difference of orientation angles of the observations o_1 and o_2 has to be between a_1 and a_2 degrees. $a_1 = a_2 = 0$ indicates that o_1 and o_2 need to be executed at the same orientation angle.

Multiple entries of the same type of constraint on the same observation(s) are taken in disjunction.

The orientation angle of an observation defines the rotation of the telescope along the line of site of the JWST detectors. JWST can only use certain orientation angles for a particular target at any given time. Thus, an orientation constraint translates into a timing constraint once we know the angle at which the other observation can or will be executed. More details on the orientation constraint can be found in (Kramer and Giuliano, 1997).

Visibility window constraint

The position of the target being observed defines the visibility windows. Ecliptic poles are visible throughout the year, while a target on the ecliptic equator (i.e. on the

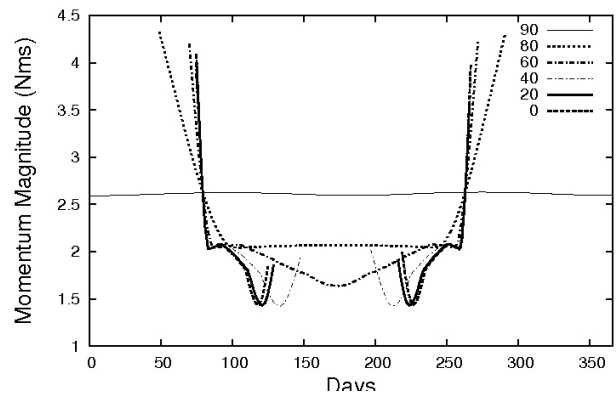


Figure 1: Momentum magnitude at different times of the year for 1 day long observations with longitude = 0 at varying latitudes. Target positions at negative latitudes produce the same momentum magnitude profile with different vector directions. Target positions at different longitudes have the same target visibility windows and momentum profile shifted to different times of the year.

same plane as the Earth's orbit) has 2 visibility windows of about 49 days each.

Momentum limit constraint

JWST requires a large solar shield to prevent sun light from contaminating the science detectors and to protect the detectors from overheating. Solar pressure on the shield creates angular momentum force that is absorbed into the reaction wheel assemblies. The wheels have a finite capacity to absorb momentum buildup as measured in Newton meters per second (Nms). The current assumption is that momentum will be dumped every 22 days during scheduled station keeping activities. Too much momentum buildup during a 22-day period will require an extra momentum dump. As momentum dumps require burning scarce fuel, too many extra dumps will shorten the lifespan of the telescope. JWST momentum presents us with a resource constraint with the following features:

- **It is time-dependent:** Momentum depends on the spacecraft position. As the spacecraft position changes over the time, the same observation will have different momentum accumulation depending on when the observation is executed. **The amount of variation in momentum usage is latitude dependent.** Targets with lower latitude have a higher variation in momentum usage (see Figure 1).
- **It is multi-dimensional:** Momentum change can be expressed in a 3-D vector. The magnitude of momentum can be calculated by calculating the length of the vector.
- **It is additive:** After a momentum dump, momentum starts with a zero vector. Scheduling adds the momentum vector for the observation at the current time to the current momentum state.

Given that momentum is additive it is possible that scheduling an observation at a particular time reduces the total momentum buildup.

Momentum buildup is determined by target position, spacecraft position, orientation angle, and the duration of task. Varying the duration of a task more or less linearly increases or decreases the magnitude of momentum with little difference in the direction of the vector. The effect of different orientation angles on momentum is ignored in our study. All momentum calculations are done at nominal orientation.

The limit of momentum accumulation depends on the actual JWST hardware design. The current limit is assumed to be 24 Nms.

Experiment Setup

Input observations and calendar

The JWST project has created a Science Operations Design Reference Mission (SODRM), which is a set of observations that closely model those expected for the actual mission. The SODRM contains observations that match the expected mission duration, target distribution, instrument configuration, and link selection. It contains the specifications for both astronomical observations as well as calibration tasks. The whole SODRM amounts to approximately 1.64 years of tasks, including time for slews and other support activities. We use a subset of the SODRM, totaling 1.2 years worth of tasks, as input to this study. It consists of 2907 tasks, including 1822 tasks that are linked to at least one other task.

The task duration varies from 70 minutes to 12 days with the median of 2.08 hours. Since the input set has a somewhat un-proportionally high number of link constraints, we have decided to also run the experiments with no link constraints and with half of the links.

The experiments are set up to schedule observations for

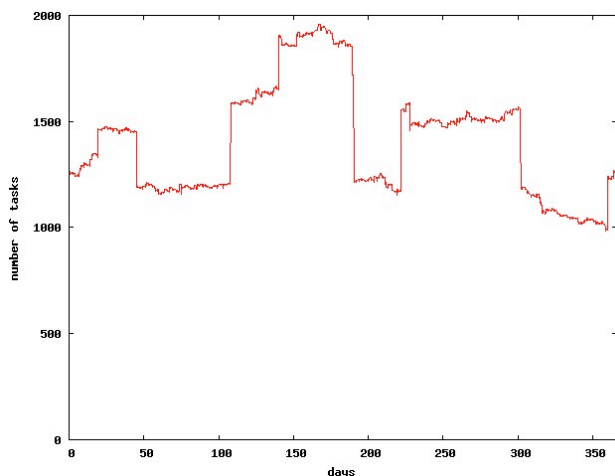


Figure 2: The number of tasks that satisfy **between** and visibility window constraints at each day.

1 year starting April 2, 2012. Figure 2 shows how many visits satisfy their absolute constraints (i.e. visibility window constraint and **between** constraint) over the year.

Scheduling constraint calculation

JWST Mission Simulator (JMS) is a software system that was created to evaluate the impact of different hardware specifications on the total mission duration. We used JMS to generate the input for SPIKE. For each task, JMS calculates the duration (= exposure time + support activity time + slew time), its visibility windows, and its momentum profile at every 3.65 days. JMS also passes user specified scheduling constraints to SPIKE input files. For our experiments, JMS used the torque table from April 2005 to calculate momentum. The slew time (i.e. the transit time between different targets) for each observation is a fixed duration of about 65 minutes. The momentum is calculated based on a fixed JWST position throughout the task including the slew time even though slew would have a different momentum profile. Since slew time is generally only a small portion of a task, where observation exposure takes the major portion of a task, we have decided this simplification is sufficient for our study.

SPIKE CSP Scheduler

The version of SPIKE used for the experiments treats scheduling as a Constraint Satisfaction Problem (CSP). Each observation is represented by a CSP variable. Each variable has a finite domain of discrete values that represent the times that the observation can schedule given its absolute constraints. The system counts conflicts on domain values during scheduling. Conflicts for a value can occur iff:

- Another observation is scheduled in that time slot
- The value would not obey a timing link from another observation
- Scheduling in that time slot would result in a momentum resource violation (only used for some experiments).

CSP SPIKE has several built-in scheduling strategies and provides templates for creating new strategies. The system supports iterative repair search algorithms. The scheduler first makes an initial guess that assigns a start time to every observation, possibly assigning observations to conflicting times. In the repair stage, SPIKE tries to reduce the number of conflicts by re-assigning the start time of conflicted tasks. At the end of the repair stage, SPIKE removes the assignments for observations with existing conflicts to produce a conflict-free schedule. For the experiments below, a simple set of gap filling routines were designed.

Time in the scheduler is represented in fixed quanta. Both the duration of tasks and their possible start and end times are expressed in this manner. Since the scheduler checks for overlaps and resource conflicts before making a commitment to a particular start time, a smaller quantum requires greater run-time memory and time. We have

picked the quantum size of 0.025 days (= 36 minutes) for the study as this enabled us to run the set of experiments in reasonable time.

The use of the quantized time introduces unavoidable gaps in schedules. For example, if a task of 80 minutes is scheduled at time t , a 22 minute gap follows the task as the quantized time after $t + 80$ is $t + 108$. We ignore such gaps in the evaluation of schedules and concentrate on the gaps that are known to the scheduler.

Schedule qualities

The JWST schedule qualities we desire are the following:

1. Low momentum accumulation.

The current operational plan is to dump momentum every 22 days during station keeping maintenance. The goal for the scheduler is to have no or very few 22-day periods that require additional momentum dumps. There are approximately sixteen 22-day periods in a year. It is acceptable for few (< 10%) of the schedules to require additional momentum dumps. These schedules could either be manually adjusted to remove the extra dumps or the additional momentum dump and resultant use of fuel could be absorbed.

2. Minimum gaps.

The JWST contract mandates 97.5% utilization of JWST; in other words, JWST should be idle for no more than 9.13 days of a year. The SODRM input set of 1.2 years worth of observations provides 20% oversubscription to fill gaps. We expect this level of oversubscription in actual operations and expect that actual operations will be able to utilize special gap filling observations.

3. Robust to observation failure

An observation can fail unexpectedly on-board for several reasons such as failure to lock onto guide stars. The on-board mechanism allows the next observation to be executed immediately after a failure. Failures could result in a different momentum profile than what was planned on the ground leading to unplanned momentum dumps.

The following experiments use momentum scheduling and gaps as evaluation criteria. The discussion section describes the interactions of autonomous operations on momentum scheduling.

Scheduling Strategy Experiments

Our goal is to find the simplest scheduling strategy that yields schedules with acceptable gaps ($\leq 2.5\%$) and momentum buildup. To this end the experiments start by examining momentum behavior with no explicit scheduling mechanism and move towards more complicated scheduling techniques. Although only a single number is shown for each experimental run, multiple runs

were performed to exercise the random components of the scheduler. The data shown are from a single run and are consistent with the data from other runs.

Baseline Experiment

Our first experiment was to see whether or not momentum buildup can be handled without using an explicit momentum based scheduling mechanism. An existing scheduling algorithm was selected that minimizes conflicts, fills gaps, and biases towards scheduling early. The scheduling algorithm was run on the SODRM data set with all the links, half the links, and no links. The total momentum accumulation is shown over time in Figure 3. Although this chart clearly shows that the baseline strategy violates the momentum limit, it does not accurately reflect how operations will handle momentum. In operations, scheduling will proceed with fixed 22-day periods. Momentum will be dumped and reset to zero at the end of each period. We want to know whether or not our techniques create schedules that require extra dumps to remain below the desired 24 Nms momentum limit. However, with no explicit model of momentum, we do not know where the 22 day boundaries occur. To account for this, we define a set of periods for each of the 22 possible start days. For each set we count the number of periods that require an extra momentum dump. We present the minimum, average and maximum number of extra momentum dumps taken over the 22 sets.

Table 1 gives a summary of the schedule quality using the baseline strategy. In each cell the top line gives the average, maximum and minimum of extra momentum dumps. The bottom line in the cell gives the gaps in days for the schedule. The baseline results in good gap performance ($\sim 1\%$ of a year). Although the gap performance is good, the momentum profile is poor in that 8 out of 16 periods in a year require extra momentum dumps. The baseline do-nothing strategy is not sufficient to handle momentum buildup.

	base
Links	7.6/9/6 4.9
Half links	8.3/10/7 3.5
No links	8.3/10/7 3.0

Table 1. Additional momentum dumps (avg/max/min) in the top row and total amount of gaps (in days) in the bottom row with baseline strategy.

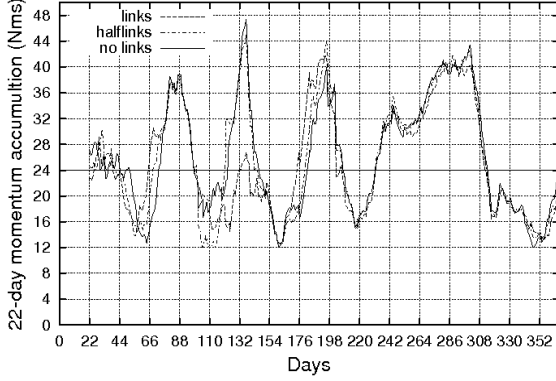


Figure 3: Momentum accumulated in the preceding 22 days with baseline strategy.

Individual Observation Techniques

In our next set of experiments, we examine whether or not we can handle momentum by biasing the scheduling of individual observations to times with lower momentum magnitude. Two mechanisms are evaluated. The first limits the constraint window for observations. The second provides a preference for times with low momentum buildup. The idea for both of these strategies is that times where the momentum vector magnitude is small are better for scheduling than times where the vector magnitude is large.

1. 50% windows (**50%**):

We reduce the constraint window for each observation by removing up to 50% of its worst momentum times. We sample all the momentum data in an observations visibility intervals and find the median momentum value. We reduce the visibility windows by subtracting sub-intervals with greater momentum magnitude than the median value. Since momentum is sampled in JMS only every 3.65 days with 2 decimal digit precision, the resulting domain is slightly larger than the 50% of the original domain. In cases where between constraints limit the domain to be less than 3.65 days, no domain reduction is taken. Other than limiting the observation window this approach uses the baseline scheduling algorithm.

2. Momentum preference (**max-pref**):

SPIKE provides a mechanism where observations can be given preference values between 0 and 1. The value at a time is a measure of goodness where 1 is most preferable and smaller values are less preferable. When the scheduler picks a start time for a task during the initial guess or repair stage, a preference is given to times with minimum momentum accumulation. More precisely, the scheduler

looks for start times that create the least number of conflicts, with the least momentum buildup.

The preference function used in the experiments is defined as follows. Let $Mom(Obs, t)$ be the magnitude of the momentum vector for observation Obs at time t and let $\max Mom(Obs) = \max(Mom(Obs, t))$ over all times t , and let $\min Mom(Obs) = \min(Mom(Obs, t))$ over all times t . The preference $\text{pref}(Obs, t)$ is given by

$$\text{pref}(Obs, t) = 1 - \frac{Mom(Obs, t) - \min Mom(Obs)}{\max Mom(Obs) - \min Mom(Obs)}$$

	50%	max-pref
Links	6.2/8/5 33.6	3.3/5/2 22.5
Half links	5.8/7/5 31.5	3.7/6/2 19.8
No links	6.1/7/5 29.2	5.3/7/4 12.1

Table 2: Additional momentum dumps (**avg/max/min**) on the top row and amount of total gaps (in days) on the bottom row with 50% and max-pref.

Table 2 shows that these approaches performed better with respect to momentum than the baseline. Although the max-pref schedules require fewer extra momentum dumps than the baseline, they still require extra momentum dumps ~20-30% of the periods. These schedules result in unacceptable gaps.

Random tie-breaker

In the experiments described above, the earliest time was picked as a tie-breaker when selecting observation start times. Observations typically have multiple minimum momentum times during a year as shown in Figure 1. With a bias towards scheduling early the scheduler prefers the first momentum minimum. This results in momentum being built up in a single direction. We investigated a tie breaking approach that randomly chooses between times with minimum conflicts and momentum build-up. The results of the experiments are shown in Table 3 and Figure 4

	base	50%	max-pref
Links	5.0 /6/4 4.2	2.6 /4/2 25.5	3.3 /4/3 17.4
Half links	4.7 /6/3 4.6	2.5 /3/1 16.6	2.4 /3/1 17.9
No links	3.9 /5/2 2.9	2.7 /4/1 8.0	2.3 /3/1 8.4

Table 3: Additional momentum dumps (**avg**/max/min) on the top row and amount of total gaps (in days) on the bottom row with random tie-breaking.

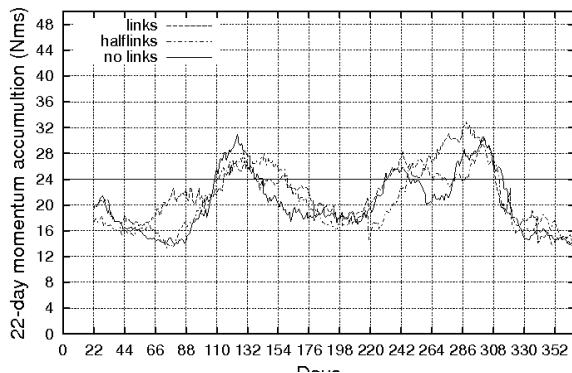


Figure 4: Momentum accumulated in the preceding 22 days for baseline strategy with random tie-breaking.

Random selection generates better schedules in terms of momentum. In addition, there is a clear trade-off between momentum scheduling and gaps. The 50% and max-pref schedules are better than the baseline with respect to momentum. However, they contain unacceptable gaps. The baseline schedule using a random tie-breaker has good gap performance but poor momentum scheduling. All of the schedules require additional momentum dumps for about 20% of the 16 periods in a year.

Resource constraint

All of the techniques considered so far attempt to handle momentum by locally controlling how each individual observation is scheduled. No explicit global model of momentum buildup is used. The approach can be paraphrased as do the best you can locally for each observation and hope that it works out globally. As seen in the data presented above, local approaches violate the 24 Nms limit and/or result in significant gaps.

CSP SPIKE has a resource mechanism designed to track single dimensional capacities. We added the ability to track a three-dimensional resource to model the accumulation of momentum in fixed 22-day periods and to add conflicts to observations for times in which they cannot be scheduled

without overflowing the momentum limit of the associated period. SPIKE uses existing conflict reduction strategies to remove conflicts due to resource violations. Other than using a resource constraint, we ran the same scheduling strategies as in the previous experiments.

The results are shown in Table 4 and Figure 5. As expected, all runs produce schedules that do not violate the momentum limit during the specified 22-day periods. Figure 5 shows that a resource constraint creates a more flat 22-day momentum accumulation throughout the year. With respect to gaps, the baseline strategy did better than 50% window or maximum-preference strategy. We believe that the baseline method allows the scheduler more flexibility in placing the tasks. Thus, the baseline explores more possibilities than other strategies while the resource constraint controls momentum accumulation.

	base	50%	max-pref
Links	5.7	29.3	21.9
Half links	5.1	22.6	18.7
No links	2.9	8.0	8.4

Table 4: Gaps (in days) in schedules created with a 24 Nms resource constraint.

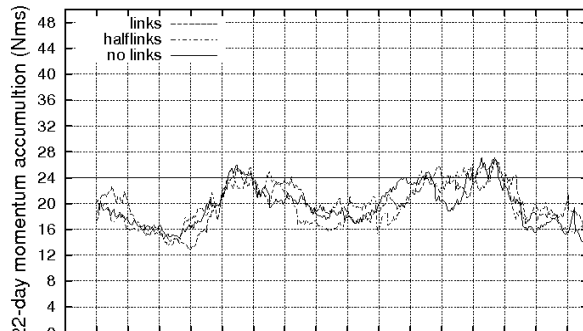


Figure 5: Momentum accumulated in the preceding 22 days for baseline strategy with a 24 Nms resource constraint. Note the lines are below 24 Nms at the end of 22-day periods used with the constraint. (i.e. days 22, 44, 66, ..)

Varying the Momentum Limit

The experiments presented so far consider a fixed momentum limit of 24 Nms. This limit could change in the future as the hardware operating procedures evolve. Additional experiments were performed to determine how changing the momentum limit would impact scheduling. We examined momentum limits of 20, 28, and 32 Nms. For each limit, we ran the max-pref and baseline strategies on the three inputs. In each case, SPIKE was able to create a schedule with no resource conflicts within the fixed 22-day periods. Tables 5-7 show the gaps for the 20, 28, and

32 Nms runs. The gaps grow larger and the runtime increases with a limit of 20 Nms (although the runtime was not explicitly tracked). Runs with a momentum limit of 16 Nms were terminated due to excessive runtime (> 1 day). Increasing the momentum limit does not substantially improve the gap behavior.

20 Nms	base	max-pref
Links	10.4	28.7
Half links	10.4	20.5
No links	5.5	17.9

28 Nms	Base	max-pref
Links	5.3	21.1
Half links	4.3	15.6
No links	3.3	8.3

32 Nms	base	max-pref
Links	5.3	24.5
Half links	4.5	15.6
No links	2.9	8.4

Tables 5-7: Gaps (in days) in scheduled created with momentum resource constraint with the resource limit of 20, 28 and 32 Nms.

We also examined at what Nms level would a non-resource constraint strategy be acceptable. Figure 4 shows the momentum profile over time for the baseline schedule using a random tie-breaker and no resource constraint. The resource level is never over 32 Nms indicating that if the resource limit were 32 Nms, then the random baseline approach would provide good momentum schedules. This strategy also produces acceptable gaps. In contrast, if the momentum limit is stricter than 24 Nms, then a non-resource constraint approach is clearly not acceptable.

Summary of scheduling results:

1. The baseline do-nothing strategy does not produce acceptable momentum results.
2. Scheduling with a bias for early values leads to poor momentum scheduling. Introducing a random tie-breaker for value selection improves momentum scheduling.
3. Using a 50% momentum constraint or a momentum preference with a random tie-breaker yields marginally acceptable momentum performance but results in unacceptable gaps.
4. Using a momentum resource constraint with the baseline strategy yields acceptable momentum performance and gaps.
5. If the momentum limit were raised to 32 Nms, then an approach using a baseline schedule with random tie-breakers and no resource constraint would be sufficient in terms of momentum and gaps.

Analysis

The experimental results show that different scheduling strategies result in different momentum outcomes, but the results do not answer how much momentum management is possible. For example, suppose we have a 10 day long observation that needs to be scheduled at a time where the contributing momentum magnitude is 20 Nms. Is it ever possible to fill the rest of the 22-day period with other tasks without violating the 24 Nms limit, or do we always have to schedule an additional momentum dump activity in such a case? We have seen in our experiments that the total momentum sometimes decreases. However, it is not clear that it is ever possible to bring the total momentum accumulation to zero or close to zero at the end of a 22-day period.

We ran a best case analytic experiment to investigate the issue. Since momentum vector directions decide if the magnitude of total momentum can decrease or not, we have looked into what the maximum possible angles are between the momentum vectors of tasks that are schedulable in any 22 day period. Note that for any two vectors \mathbf{v}_1 and \mathbf{v}_2 , $|\mathbf{v}_1 + \mathbf{v}_2| \leq |\mathbf{v}_1|$ is possible only if $\theta \geq 90$ where θ is the angle between the vectors. Also, for $\theta \geq 120$, $|\mathbf{v}_1 + \mathbf{v}_2| \leq \max\{|\mathbf{v}_1|, |\mathbf{v}_2|\}$ always holds. In other words, if the maximum angle between the momentum vectors of every pair of schedulable tasks in the same 22-period is less than 90 degrees, then it is not possible to reduce the total momentum buildup during that period. On the other hand, the effect of a momentum contribution from a task can be reduced if we find another task with a momentum vector that is greater than 120 degrees apart. We created an input with 206 three hour-long observations that systematically covers the sky at every 30 degrees in longitude and every 10 degrees in latitude. Using JMS, we created a SPIKE input file where the momentum data are sampled every day. Then using the same set of sixteen 22-day periods as used in the resource constraint experiments, we calculated the maximum possible angle between each pair of tasks schedulable within the 22-day period. The results are the following:

- a. For some high latitude tasks, there exists another task in at least one of the 22-day periods where the angle between the momentum vectors is 180 degrees.
- b. For any task, there exists another task in at least one of the 22-day periods where the angle between the momentum vectors is at least 166 degrees.
- c. For any task in any 22-day period where it has a visibility window, there exists another task where the angle between the momentum vectors is at least 130 degrees.

The findings seem to suggest that for any task at any time, it may be possible to find other tasks that nullify the effect of its momentum (at 180 degrees) or at least significantly

lessen it (at 130 degrees). However, we also found that the following holds:

- d. Pairs of tasks with greater momentum vector angles typically have targets very far apart; opposite hemispheres and about 120 degrees difference in longitude.
- e. The 22-day period in which the momentum for a task can best be offset by another task is the period where both tasks have their greatest magnitude in momentum (i.e. on the edges of Figure 1).

Item (d) suggests that attempting to make momentum sum to zero may take time away from science, as the telescope will require long slews. Since our scheduling experiments do not have a slew model that reflects position change, we cannot assess this effect. The finding does explain the reason for the baseline strategy doing better than the max-pref strategy when using a resource constraint; it is more likely to put mixed tasks (targets) in 22-day periods, resulting in more varied momentum vector directions. Item (e) suggests that trying to balance the momentum vector direction may not be truly effective since the maximum momentum magnitude can be as much as 2.8 times the minimum. Also, scheduling a task at the maximum momentum time creates a less robust schedule with respect to unexpected task failure. Suppose a schedule balances momentum by scheduling observations with complementary vectors. If one of the pair fails, then momentum will build up in the direction of the other observation. Since complementary pairs necessarily have high magnitudes, a momentum violation is likely to result from such failure and an unplanned momentum dump may be required.

We repeated the same experiment using the 50% window constraint. As anticipated, we found smaller maximum angles. The maximum possible angle difference is 164 degrees (compared with 180 degrees above) for some tasks and at least 133 degrees for any task (compared with 166 degrees). If we limit a task to a particular period, then the maximum angle difference possible can be as small as 117 degrees (compared with 130 degrees). Item (d) still holds true in the run with 50% window, indicating greater slew time is required to achieve momentum balancing.

These findings suggest that momentum can be managed greatly through scheduling if given a wide distribution of input observations. However, the robustness of the schedule and the availability of science time may be compromised.

Conclusions and Future Work

We evaluated JWST scheduling techniques ranging from simple constraints on single observations to complex modeling of a momentum resource. The results show that for the expected hardware limit, an explicit resource

constraint is needed to obtain efficient schedules that do not violate momentum resource limits. In addition, we performed analytic experiments designed to determine whether or not it is possible to create schedules with little or no total momentum buildup. These results show that it is possible to find observations whose momentum vectors either fully or partially negate each other. However, there are two mitigating factors. First, negating pairs would require substantial slew time leaving less time for science. Second, these pairs of observations occur at the maximum momentum magnitude point for the observations (i.e. on the edges of Figure 1). If one of these observations fails, then the total momentum buildup could be very high. Whether or not balancing techniques are effective will be the subject of future investigations.

There have been several studies on using resource constraints during scheduling. (Laborie, 2003) considers one-dimensional resources where the amount of usage does not vary with the task start time. We demonstrate that a one-dimensional resource constraint mechanism can be extended to effectively handle three-dimensional resources where resource usage varies over time.

Changes to the JWST hardware design can significantly affect observation momentum profiles. Thus, the results we have reached here may not hold true in the actual JWST scheduling system. However, the tools and methodologies we have developed will help us to analyze alternative hardware configurations, and to design a better operational scheduler.

Acknowledgements

Larry Petro directed the initial momentum experiments. We thank the reviewers as well as Robert Hawkins for their comments. We also would like to thank all Spike stakeholders past and present for their efforts.

References

- Calvani, H.M., et al., 2004. The evolution of the FUSE Spike Long Range Planning System. In Proceedings of the 4th International Workshop on Planning and Scheduling for Space. Darmstadt Germany.
- Johnston, M. and Miller, G. 1994. Spike: Intelligent Scheduling of Hubble Space Telescope Observations. In Zweben M. and Fox M. eds. *Intelligent Scheduling*, 391-422. Morgan-Kaufmann.
- Kramer, L. 2000. Generating a Long Range Plan for a New Class of Astronomical Observatories. In *Proceedings of the 2nd NASA Workshop on Planning and Scheduling for Space*.
- Kramer, L. and Giuliano, M. 1997. Reasoning About and Scheduling Linked HST Observations with SPIKE. In *Proceedings of the NASA Workshop on Planning and Scheduling for Space*. Oxnard, CA.

Krueger, T. 2004. APT: Hubble Space Telescope's Proposal Preparation Tool, In *Astronomical Data Analysis Software and Systems XIV*.

Laborie, P. 2003. Algorithms for Propagating Resource Constraints in AI Planning and Scheduling: Existing Approaches and New Results. *Artif. Intell.* 143(2):151--188.

Sasaki, T., et al., 2000. Observation scheduling scheme for Subaru telescope, in *SPIE*.