Multi-Objective Evolutionary Algorithms for Scheduling the James Webb Space Telescope

Mark E. Giuliano
Space Telescope Science Institute
3700 San Martin Drive, Baltimore MD 21218
giuliano @stsci.edu

Mark D. Johnston
Jet Propulsion Laboratory/California Inst. Of Technology
4800 Oak Grove Drive, Pasadena CA 91109
mark.d.johnston @jpl.nasa.gov

Abstract

Effective scheduling of the James Webb Space Telescope (JWST) requires managing the trade-off between multiple scheduling criteria including minimizing unscheduled time, angular momentum build-up, and the number of observations that miss their last opportunity to schedule. Previous studies examined momentum management and wasted space and showed that effective JWST scheduling requires modeling momentum as a resource that is three-dimensional, where activities can either produce or consume resources depending on when they are scheduled. We enrich the scheduling model by adding the ability to schedule JWST at different spacecraft roll angles and show that this ability has a strong impact on managing momentum. A series of multi-objective evolutionary algorithms are developed which incorporate different techniques to search the enriched domain. The algorithms are empirically evaluated showing that the best solutions are generated by the approach that evaluates the least number of candidate solutions.

Introduction

Effective scheduling of space based astronomy missions requires the ability to make trade-offs between competing mission objectives. A typical mission includes many objectives such as increasing time on target, minimizing use of consumables, minimizing the use of critical mechanisms, and preferring the highest priority science first. These objectives are often competing in that improving one objective means making another worse. The objectives also have different constituents lobbying for them. For example, the mission science community may have different needs from the engineering community. Traditional scheduling optimization techniques are generally based on a single objective that combines all criteria into a single value, often by weighting the values of the individual objectives. However, this necessarily loses information about the individual components of the objective, and pre-determines the tradeoff among them. Multi-objective scheduling techniques allow the retention of separate objective components and thus for explicit visibility into trade-offs.

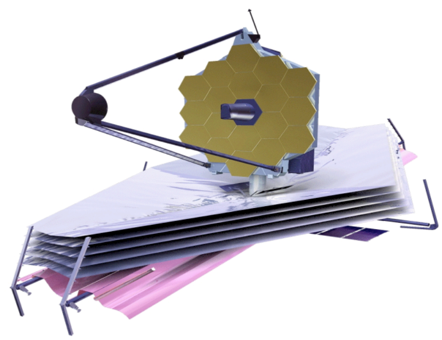

Figure 1. James Webb Space Telescope

Multi-objective scheduling is a good match for the James Webb Space Telescope (JWST). In addition to typical scheduling objectives such as minimizing schedule gaps and minimizing the number of observations that miss their last scheduling opportunity, JWST scheduling requires the ability to minimize momentum build-up during scheduling. 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 angular momentum to accumulate in the spacecraft's reaction wheel assemblies as measured in Newton meter seconds (Nms). The wheels have a limited capacity to store momentum, and stored momentum must be dumped using spacecraft thrusters. The resulting use 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. The JWST momentum resource constraint has several interesting features:

- the model is intrinsically three dimensional

- resource consumption for an observation varies over time in non-linear manner
- resource consumption is vector additive in nature — scheduling an observation at a particular time can either add to or subtract from the overall accumulation
- momentum provides both a hard constraint due to a limited capacity, and a preference to consume as little resource as possible.

These features are different from the types of resources covered in the planning and scheduling literature (Laborie 2003; Policella *et al*. 2004) where activities consume and release a constant capacity. In particular, the non-linearity of the domain prevents us from employing techniques commonly used to handle resource constraints.

Previous studies on JWST scheduling (Rager and Giuliano 2006, Giuliano, et al 2007) demonstrated effective heuristics for scheduling JWST within a simplified spacecraft model. In this paper we enrich the scheduling model for JWST by introducing a spacecraft roll decision variable and show that controlling spacecraft roll is a major factor in limiting momentum build-up. Scheduling systems can incorporate multiple decision variables in many ways, ranging from optimizing all variables at once, to having separate searches for each variable. The choice of algorithm impacts the portion of the search tree explored and the number of schedule candidates evaluated. We examine different techniques for incorporating a roll search into a multi-objective evolutionary scheduling system, as well as heuristics for scheduling. The resulting algorithms are empirically evaluated based on experimental runs using a set of JWST test observations showing that the technique that evaluates the most solutions does not necessarily create the best schedules.

JWST Mission Operations

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. Components of the mission are under construction and launch is planned for 2013. JWST will have infrared sensitive detectors and a 6.5-meter primary mirror designed to look through dust clouds to see earliest formation of stars and planets. The telescope will have a lifetime of 5 to 10 years and will be placed in 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. Programs will be submitted to the JWST Science Operations Center (SOC) and will be scheduled by SOC staff using a two phase scheduling process similar to the process used for the Hubble Space Telescope (Giuliano 1998). In the first phase, a long range plan assigns observations to overlapping least commitment plan windows that are nominally 60 days long. Plan windows are a subset of an observation's schedulable windows and represent a best effort commitment to schedule within the window. In the second phase, successive short-term schedules are created for 22 day upload periods. The short-term scheduler uses plan windows to drive the creation of efficient telescope schedules. This two phase process allows a separation of concerns in the scheduling process: Plan windows globally balance resources, are stable with respect to schedule changes, and provide observers with a time window so they can plan their data reduction activities. Short-term schedules provide efficient fine grained schedules to the telescope, handle slews between observations, and provide schedules robust to execution failure.

JWST Scheduling Constraints

A scheduling system for JWST has to satisfy several types of constraints on observations. First, an observation has to satisfy all requirements defined by the user. These include the ability to specify time windows for observations, to link observations via precedence or grouping relationships with offsets, and to link observations via roll constraints.

Secondly, an astronomical target can be observed by JWST only at certain times of the year determined by the location of JWST relative to the sun and the target. We call such time intervals *visibility windows*. The celestial 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 same plane as the Earth's orbit) has two visibility windows of about 49 days each.

Thirdly, schedules must satisfy limits on momentum accumulation. The current assumption is that stored angular momentum will be dumped every 22 days during regularly scheduled station-keeping activities. Momentum buildup during a 22-day period over a 24 Nms limit will require an extra momentum dump. As momentum dumps require burning scarce fuel, extra dumps will shorten the lifespan of the telescope. It is also preferable to minimize the amount of momentum dumped during the regularly scheduled dumps.

Momentum accumulation can be controlled by adjusting the spacecraft roll angle. At any pointing within the field of regard, JWST can roll ±5° from the normal angle without violating spacecraft constraints. Rolling the telescope for an observation impacts the angle that the solar pressure asserts on the sunshield, thus affecting the momentum buildup for the observation.

Schedule qualities

The JWST schedule qualities we desire are the following:

1. Minimize schedule gaps. The JWST contract mandates 97.5% scheduling efficiency. The input set (described below) of 1.2 years worth of observations provides 20% oversubscription to fill gaps in a one year schedule. We expect this level of oversubscription in operations and expect that operations will be able to utilize special gap filling observations.

2. Minimize 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. In addition to the 24 Nms momentum limit, it is preferable to lower the amount of momentum dumped during scheduled station keeping maintenance as that reduces the amount of non-renewable fuel to be used.

3. Minimize dropped observations. The JWST scheduling process first assigns *plan windows* to observations during long range planning. Plan windows are a subset of an observation's constraint window and are created to balance global resources while informing the astronomer when to plan data reduction activities. Missing a plan window can disrupt the resource balancing, break the handling of linked observations and disrupt the plans of the astronomer end user. In the worst case, a missed observation cannot be performed later in the year and has to either be reworked or scheduled in the next year.

Input Observation Set

The JWST project has created a Science Operations Design Reference Mission (SODRM), which is a set of observations that closely match the expected mission duration, target distribution, instrument configuration, and constraint selection. It contains the specifications for both astronomical observations as well as calibration observations. The entire SODRM amounts to approximately 1.64 years of observations, including time for slews and other support activities. To allow us to compare with the previous studies we use the same subset of the SODRM, totaling 1.2 years worth of observations, as input to this study. It consists of 2907 observations, including 1822 observations that are linked to at least one other observation. Observation duration varies from 70 minutes to 12 days with a median of 2.08 hours.

Scheduling constraints for observations in the SODRM were calculated using the JWST Mission Simulator (JMS). For each observation, 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 such as links between observations to SPIKE.

To model momentum at different rolls JMS routines were modified to provide momentum data at the normal angle, +/-1.5 degrees, and +/- 3.0 degrees from normal. Although larger rolls from nominal are legal they were not considered as scheduling close to the legal limit could create schedules that fail during on-board execution.

The momentum routines were run using updated spacecraft parameters that reflect best-case momentum build-up. As a side effect, the momentum levels in the experimental results cannot be directly compared to the results presented in the previous studies.

Verifying Previous Results

Previous work on JWST scheduling (Giuliano et al. 2007) developed a momentum balancing heuristic and showed that it was effective in both long range planning and short-term scheduling. Using the new roll data we performed experiments to revalidate the previous results and to determine the potential impact of using roll to impact momentum build-up. The experimental setup from (Giuliano et al. 2007) was repeated using the new roll tables. The new results duplicate the previous experiments showing that the heuristic was effective in both long range planning and short-term scheduling reducing momentum build up by up to 20 percent.

The Impact of Roll on Momentum Build-up

Next the short-term scheduling experiments were repeated with a roll search being performed after each schedule was created. The roll search assigned all scheduled observations a random legal roll 100 times and picked the schedule with the lowest momentum build-up. The results show that even a random roll search has a major impact on momentum build-up. Using the momentum heuristic reduces build-up by 20%. Using a very simple roll search resulted in 80% reduction in total momentum build-up.

Based on these results our next set of experiments concentrated on strategies for integrating a roll search into short-term scheduling. The heuristic LRP generated to perform the experiments summarized above was used to drive the creation of schedules for three successive 22-day resource bins. The experiments explore heuristics and alternative methods for incorporating a roll search into an evolutionary algorithm based application architecture.

Evolutionary Algorithms

A multi-objective optimization problem to minimize M objectives subject to K constraints can be stated as follows:

$$\text{minimize: } \left\{ f_i(\vec{x}) \right\},\quad i = 1 \ldots M$$

$$\text{subject to: } \left\{ g_j(\vec{x}) \right\}^T \leq 0,\quad j = 1 \ldots K$$

Here $\vec{x}$ represents a vector in decision space of dimension D. A solution is called *Pareto optimal* when no improvement can be made to one objective that does not make worse at least one other objective. The set of Pareto optimal solutions is called the *Pareto frontier*. What we seek as a solution to the multi-objective optimization problem is a good approximation to the Pareto frontier. Two important characteristics of a good solution technique are convergence to the Pareto frontier, and diversity so as to sample the frontier as fully as possible.

We have adopted an evolutionary algorithm approach (Deb 2001, Collete and Siarry 2003, Abraham, Jain, and Goldberg 2005) to JWST scheduling. Among techniques developed to solve multi-objective optimization problems, evolutionary algorithms have become popular for a variety of reasons. They have been shown effective on a wide range of problems and are capable of dealing with objectives that are not mathematically well behaved (e.g. discontinuous, non-differentiable). By maintaining a population of solutions they are capable of representing the entire Pareto frontier at any stage. They also lend themselves to parallelization, which is an important performance consideration for large problems.

For this study we have utilized one particular variant called *Generalized Differential Evolution 3*, or GDE3 (Kukkonen and Lampinen 2005) which has been previously used in multi-objective scheduling in a space network application (Johnston 2006, 2008). This technique is based on *Differential Evolution*, a single objective evolutionary algorithm for real-valued decision spaces (Price, Storn, and Lampinen 2005). GDE3 makes use of concepts pioneered in the algorithm NSGA II (Deb et al. 2002), including:

- *non-dominated* sorting of the population into ranks, such that members of rank *n* dominate members of all ranks >*n*, where rank 1 members constitute the non-dominated set, i.e. the current approximation to the Pareto frontier
- *crowding distance* is used as a secondary discriminator on members of the same rank: members in crowded regions of the population are scored lower, so the surviving members after selection have greater diversity. This helps prevent premature convergence of the population to a small portion of the Pareto frontier
- population members are compared with a *domination* or *constraint-domination* relation — the latter allows for domination comparisons even when constraints are violated

GDE3 operates as follows to evolve the population of size *N* from one generation to the next:

1. For each parent member of the population $\vec{x_i}$, select three distinct population members $\vec{x_{r_1}}$, $\vec{x_{r_2}}$, $\vec{x_{r_3}}$, all different and different from the parent
2. Calculate a trial vector $\vec{y_i} = \vec{x_{r_1}} + F \cdot (\vec{x_{r_2}} + \vec{x_{r_3}})$, where F is a scaling factor
3. Modify the trial vector by binary crossover with the parent with probability *CR*. The result is compared with the parent: if either the parent or trial vector dominates the other, then that vector is selected; if neither dominate, then both are selected and the population size is reduced via the non-dominated sorting and crowding distance comparisons.

System Architecture

The system architecture for the experiments integrates existing components (see Figure 2). The Java-based GDE3 component is the multi-objective evolutionary algorithm driver (Johnston 2006). The Lisp based SPIKE system has a model of the JWST scheduling domain. The evolver component sends SPIKE decision vectors that are used to create schedules and to return objective function values. The systems communicate via a socket connection.

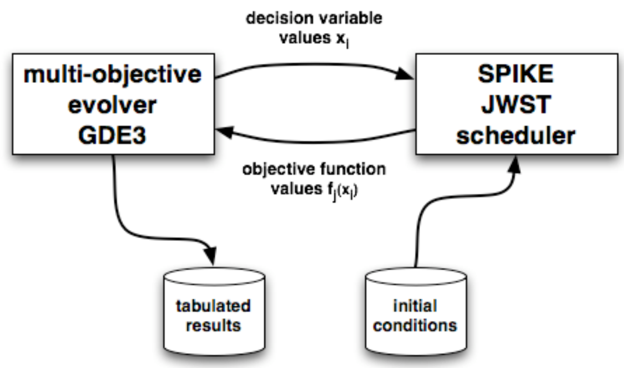

Figure 2. System Architecture

SPIKE (Johnston and Miller, 1994) is a planning and scheduling tool kit that was created for use on the Hubble Space Telescope and has been used for multiple orbital and ground based astronomical missions including FUSE (Calvani 2004), Chandra, Subaru (Sasaki 2000), and Spitzer (Kramer 2000). 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 all selected 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 observations. At the end of the repair stage, SPIKE removes the assignments for observations with existing conflicts to produce a conflict-free schedule. A simple set of gap filling routines were designed for the experiments below.

There are several potential strategies that could be adopted for integrating these two components. At one extreme, the multi-objective solver could incorporate a model of the scheduling domain in some detail; at the other, all of the detailed scheduling domain knowledge would be retained strictly in the SPIKE component. We opted for the latter strategy in the work described here: this shows the relative ease with which a multi-objective solver can be combined with a scheduler that was not designed with this in mind, and also prevents the duplication of constraint checking and objective evaluation algorithms that would otherwise be required.

Evaluating the Evolutionary Algorithm Approach

Experiments were performed to evaluate the evolutionary algorithm approach to JWST scheduling. An initial implementation utilized two layers of evolutionary search. A top-level algorithm repeatedly calls SPIKE to create and evaluate schedules. A lower level embedded loop searches the roll space for each schedule created.

The top-level evolutionary algorithm creates, evaluates and evolves population members that are represented as decision vectors with values in [0,1]. The vector contains a value for each observation in the long range plan for the 22-day bin. SPIKE uses the vector to drive the creation and evaluation of a schedule. The observations schedulable in the 22-day bin are sorted by the vector value and SPIKE schedules the first 22-days worth of the observations on the sorted list. SPIKE then removes conflicts and applies gap-filling algorithms that can schedule any observation schedulable in the bin. For each schedule, SPIKE returns the gap and dropped observation metric values. The evolver runs a population of 20 candidates for 20 generations calling SPIKE 400 times to create a schedule.

After each schedule is produced by SPIKE a roll search is performed that explores the possible roll assignments for the observations on the schedule, searching for roll angle assignments that minimize the total momentum build-up. An evolutionary roll algorithm was compared to a search that randomly assigns orientation values. In the evolutionary algorithm approach each decision vector contains a 0-1 element for each observation that was scheduled in the top-level search. SPIKE routines map the vector values into legal offsets from nominal and evaluate the total momentum usage of the schedule. The evolutionary roll search also runs a population of 20 candidates for 20 generations. After the roll search finished, the best momentum value is given to the top-layer evolver as the momentum metric.

In summary, the search is divided into two layers. The top layer repeatedly creates and evolves schedules. For each schedule created, a second embedded layer searches over the possible roll assignments for the observations in the schedule. The top-level search creates and evaluates 400=20x20 schedules. Likewise, the lower level search creates and evaluates 400 roll assignments for each of the 400 schedules. This gives a total of 160,000 schedules evaluated.

As originally defined, the GDE3 algorithm starts with a randomized population, in order to start with a wide range of candidates sampling a large portion of objective space. This helps ensure that the evolved population generates a broad sampling of the Pareto frontier. Since the method is elitist, non-dominated solutions will remain in the population from one generation to the next. While randomness in the initial population does indeed help with diversity in the final population, it can lead to unnecessarily many solution evaluations. Motivated by this, we experimented with biasing or "seeding" the initial population with a mixture of heuristic-generated solutions. We sought a balance between a randomized population, and one with a starter set of candidates that help speed the evolution.

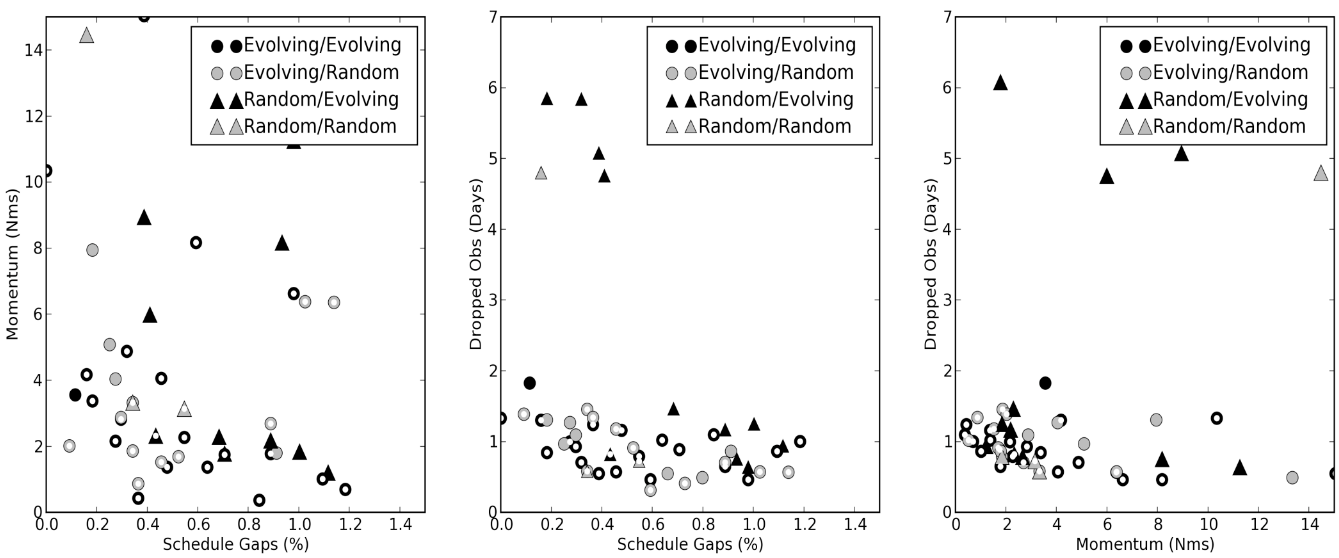

Figure 3a-c. Compares evolutionary versus random search by plotting the Pareto optimal surface generated for a 22-day schedule for each pair of metrics. The hollow data points are those in the Pareto Optimal surface obtained by combining the schedules from all four scheduling approaches.

A simple heuristic was designed that helps to minimize the number of dropped observations. By default the evolutionary algorithm generates a set of random solution vectors for the initial population. The heuristic biases the decision vector so that values for "must-go" observations were randomly distributed in the range 0-0.5, while values for non "must-go" visits were distributed in the range 0.5-1.0. Using random values in two different ranges allows for diversity while ensuring that the underlying scheduling algorithm will include the "must-go" observations in the schedules created for the initial population.

The effectiveness of GDE3 search is compared with random search algorithms that repeatedly populate decision vectors with randomly generated values. Two variables are considered in these experiments:

- Whether or not the top-level search uses a GDE3 versus random search.
- Whether or not the embedded roll search uses a GDE3 versus random search.

Varying the two variables gives four combinations, with results summarized in Figures 3a-c.

The experiments were run on three successive 22-day schedule bins. As the results were consistent across bins, the experimental results from bin3 were selected for display. The results for each experimental combination are shown in Figures 3a-c. Each subfigure graphs a pair of criteria values for solutions on the Pareto frontier. The results show that using differential evolution in both the top-level and the roll search results in the best schedules. Using the evolving search at the top-level and a random roll search is the next best and the other two approaches result in significantly worse schedules. To further quantify the benefits we combined the metric results from all four runs and created a combined Pareto-optimal frontier (shown as hollow points in Figures 3a-c). We then determined which scheduling approach contributed most to the combined frontier. Using the evolver in both steps produced 56% of the combined frontier. Using the evolver just at the top level produced 35% of the combined frontier with the two random approaches producing 3% and 6% respectively.

An additional experiment was performed to further verify the effectiveness of the differential evolution algorithm over a random search when searching for roll assignments. Forty generations of evolutionary roll search were run on 200 schedules and the resulting best momentum for each schedule was stored. For each of the schedules, a random roll search was performed and the number of random assignments required to obtain a momentum that is as good as the corresponding evolving search was determined. Each of the evolving roll searches evaluates 40 generations times 20 population size = 800 roll assignments. The random search times out after 25,000 random roll assignments. The average number of random roll assignments was about 16,000. However, approximately half of the random searches time out. So the evolving roll search is at least 20 times more effective than a random search in terms of the amount of additional work required to find as good a solution.

Alternative Approaches to Integrating Roll Search

The experiments presented above show that an evolving roll search is effective at reducing momentum buildup. While running the experiments we informally observed that evaluating a roll decision vector is about an order of magnitude quicker than evaluating a schedule vector. This imbalance suggested that we should explore alternatives to the embedded roll search approach.

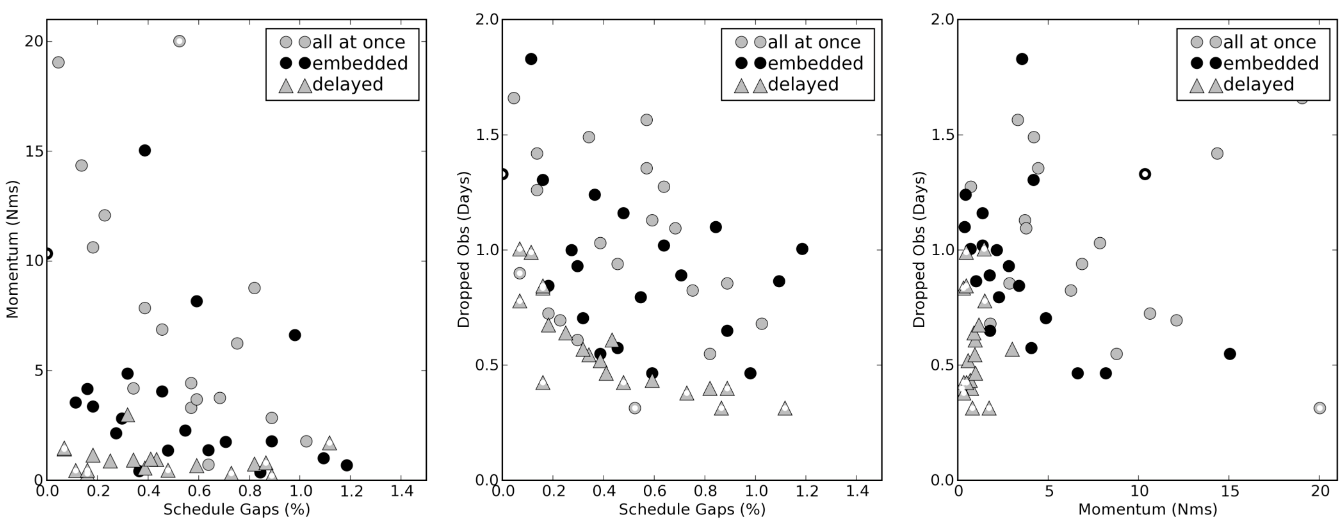

Figure 4a-c Compares three alternative approaches to integrating a roll search by plotting the Pareto optimal surface generated for a 22-day schedule for each pair of metrics. The hollow data points are those in the Pareto Optimal surface obtained by combining the schedules from the three approaches to integrating a roll search.

The nature of the roll objective, determined by a set of roll angle control variables that can be applied to any existing schedule, suggests that the overall optimization problem may be at least in part decomposable with respect to task ordering and roll. Reducing the dimensionality of the problem objective space can lead to major savings in runtime, and so we have conducted experiments to explore this. An advantage of pursuing this decomposition is that, for a fixed population size, filling out a 2D vs. a 3D Pareto frontier will show significantly better sampling.

Two additional approaches to integrating a roll search into the overall search process were considered. In the first, we performed a single level evolutionary search that attempted to optimize both sets of decision variables at once: we call this the “all-at-once” method. The decision vector had two entries for each observation schedulable in the bin: one entry gave the sort order and the other the roll assignment. The second alternative was to delay the roll search until after a good set of candidate solutions were generated. The evolutionary algorithm was run just considering wasted space and minimizing dropped observations. After the search completed, a roll search was performed on all the population members that were active in the last generation of the search. This second approach effectively decomposed the problem into (1) a two-dimensional multi-objective search, followed by (2) a single-objective roll optimization for each of the members of the final generation.

One way to compare the algorithms is to have each approach evaluate the same number of candidate solutions. This approach is not used as the time to evaluate a schedule is at least an order of magnitude larger than the time to evaluate a roll assignment. Instead we adjusted the number of generations expanded in the different search approaches so that each search had more or less the same runtime. The parameters were set as shown in Table 1. (all population sizes are set at 20):

	Schedule and Roll Search	Candidates Evaluated
Embedded	20 generations top level * 20 generations roll search	160,000 = (20*20)*(20*20)
All-At-Once	150 generations combined search	3000 = 150 * 20
Delayed	40 generations schedule + 100 generations roll	2,800 = 40*20 + 100*20

Table 1: Population members evaluated in different roll scheduling approaches

The three alternate approaches were evaluated by running them on three successive 22-day schedule bins. As in the experiments described above the results are consistent across the three bins. Figures 4a-c give the results for bin3. The figures plot each pair of schedule metrics against each other. In all three plots the delayed roll search schedules dominate those from the other approaches. As in the previous section we merged the Pareto frontiers from the three different approaches and created a single combined Pareto frontier (shown as hollow points in figures 4a-c). The delayed approach contributes 79% to this surface, while the embedded and all-at-once approaches contribute 14% and 7%, respectively. The delayed approach decomposes the problem into separate searches for schedules and roll assignment. Decomposing the search allows a deeper schedule search than the embedded approach and a more focused two-dimension search than the all at once approach. Increasing the dimensionality of a search generally requires a power of two more generations and a corresponding increase in population size in order to get results that fairly sample the Pareto frontier. The delayed approach allows for a deep roll search over a single dimension for the 20 schedules in the final generation. The resulting momentum is lower than in the approach where a shallow roll search is done for each generated schedule.

Decomposing the search into separate optimization steps is effective as the search spaces do not interact. The embedded versus the all-at-once approach results in better momentum build-up. However, the all-at-once approach is competitive when considering schedule gaps versus dropped observations (see Figure 4b). The all-at-once schedules fill in and extend the Pareto-frontier developed in the delayed roll search approach. Having a deep 150 generation search is effective when optimizing schedule gaps and dropped observations. The fact that the approach that generated the least number of candidate schedules has the overall best results shows the importance of choosing the algorithmic approach which best matches the application domain.

Conclusion

A JWST scheduling model that adds the ability to alter momentum usage based on controlling spacecraft roll was presented. The results of previous studies were reconfirmed with updated momentum data, and the importance of controlling roll to adjust momentum usage was demonstrated. Evolutionary algorithms for short-term scheduling a subset of the JWST design reference mission were presented and shown to be more effective than a randomized algorithm at optimizing the three important objectives — wasted time, accumulated angular momentum, and observations that miss their scheduling window. Finally, three alternate approaches to integrating a roll assignment search into the scheduling process were presented ranging from optimizing the two sets of decision variables all at once to decomposing the problem into two separate search processes. Experimental results were presented showing that for this domain, the best approach is to decompose the problem into a schedule search (minimizing wasted space and missed observations) followed by a roll search (to minimize angular momentum buildup).

The new results show substantially lowered momentum usage compared to the previous studies from 2006 and 2007. This is a result of the best-case spacecraft parameters in calculating momentum usage and the ability to control momentum usage by adjusting spacecraft roll angle. Although the new results show low absolute momentum build-up, they should not be taken to indicate that momentum scheduling is not a serious scheduling concern: for example, changes in the JWST sun shield design could increase momentum build up by an order of magnitude.

The evolutionary multi-objective approach has advantages that go beyond optimization, however. By generating a population of candidate Pareto-optimal solutions, the end users of the schedule are given new and valuable insight into tradeoffs available to them. These tradeoffs are often the essence of real-world scheduling, in that generating "point" solutions do not address user's questions and concerns about finding the "best" schedule for all stakeholders. By representing multiple objectives, it is possible to engage multiple participants in the scheduling process, and this is one of our directions of future research.

It is also worth pointing out that a key element of the multi-objective approach, that of keeping objectives separate until a selection decision is required, has a further virtue of not "hiding" some solutions. For example, while a linear combination of objectives with adjustable coefficients could be used to identify points on the Pareto frontier, this only works for convex frontier surfaces. There is no assurance that in realistic nonlinear problems, the surface is in fact convex.

In addition to investigating multi-participant scheduling, we plan to further extend this research in several other directions:

- The most time consuming computation in the experiments reported here is the evaluation of the short-term schedule: we plan to investigate the use of parallelized schedule evaluations, possibly using grid techniques, to speed up the overall algorithm.
- The significant improvements shown by "biasing" the initial population with heuristically generated seed solutions and the "reducer" heuristics given in (Giuliano et al 2007) suggest that investigation of additional heuristics would be worthwhile.

Design of the operational scheduling system for JWST started in the fall of 2007 and is on-going. One of the first components to be coded will be the routines that calculate JWST scheduling constraints. By integrating our system with these components we can incorporate the latest scheduling and momentum restrictions. A continuing challenge will be to integrate the algorithms developed above into the operational system. This requires adding features not included in the above experiments such as:

- The insertion of momentum dumps if momentum buildup violates the 24 Nms limit.
- Modeling slew durations between observations. These experiments assume a constant slew duration.
- The addition of a schedule metric that measures the robustness of schedules uploaded to JWST when observations fail.

Acknowledgments

Thanks to Reiko Rager, and Rob Hawkins for their careful reading of the paper. Thanks to Nazma Ferdous for help with the figures. Finally we would like to thank the ICAPS reviewers for their comments.

The research described in this paper was carried out at the Space Telescope Science Institute under the NASA Applied Information Systems Research Program grant number NNX07AV67G, and at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

Abraham, A., L. Jain and R. Goldberg (2005). *Evolutionary Multiobjective Optimization*. Berlin, Springer.

Calvani, H.M., Berman, A.F. Blair W.P. Caplinger, J.R. England, M.N. Roberts, B.A. Hawkins, R. Ferdous, N. and Krueger A.P., 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, 25-31.

Collette, Y. and P. Siarry (2003). *Multiobjective Optimization*. Berlin, Springer.

Deb, K., Multi-Objective Optimization Using Evolutionary Algorithms. (2001), New York: John Wiley & Sons.

Deb, K., A. Pratap, S. Agrawal and T. Meyarivan (2002). "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." IEEE Transactions on Evolutionary Computation **6**(2): 182-197.

Giuliano, M. 1998. Achieving Stable Observing Schedules in an Unstable World. In *Astronomical Data Analysis Software and Systems VII*. 271-274.

Giuliano, M., Rager, R., Ferdous, N. (2007) Towards a Heuristic for Scheduling the James Webb Space Telescope. In Proceedings of the International Conference on Automated Planning and Scheduling. Providence, Rhode Island. 160-167.

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.

Johnston, M. D. (2006). "Multi-Objective Scheduling for NASA's Deep Space Network Array." International Workshop on Planning and Scheduling for Space

(IWPSS-06). Baltimore, MD, Space Telescope Science Institute.

Johnston, M. D. (2008). "An Evolutionary Algorithm Approach to Multi-Objective Scheduling of Space Network Communications." *International Journal of Intelligent Automation and Soft Computing*: in press.

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*.

Kukkonen, S. and J. Lampinen (2005). "GDE3: The Third Evolution Step of Generalized Differential Evolution." The 2005 Congress on Evolutionary Computation.

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

Policella, N., Oddi, A., Smith S.F., and Cesta, A. Generating Robust Partial Order Schedules. In *Proc of CP 2004,*Lecture Notes on Computer Science (LNCS) Vol. 3258, pp. 496-511, M. Wallace (Ed.), Springer, 2004.

Price, K., R. Storn and J. Lampinen (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Springer.

Rager, R. and Giuliano, M. 2006. Evaluating Scheduling Strategies for JWST Momentum Management. In *Proceedings of the 5th International Workshop on Planning and Scheduling for Space, 235-243*.

Sasaki, T., Kosugi, G., Kawai, J., Kusomoto, T., Koura, N., Kramer, L., Krueger, A., Miller, G. 2000. Observation scheduling scheme for the Subaru Telescope, In *Proceedings of Advanced Telescope and Instrumentation Control Software*, ed. Hilton and Lewis, *SPIE Vol 4009, 350-354*.

ICAPS 2008
Proceedings

This paper was published in the *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling,* edited by Jussi Rintanen, Berhard Nebel, J. Christopher Beck, and Eric Hansen (Menlo Park, California: AAAI Press).

The 2008 ICAPS conference was held in Sydney, Australia, 14–18 September, 2008.