MUSE: THE MULTI-USER SCHEDULING ENVIRONMENT FOR MULTI-OBJECTIVE SCHEDULING OF SPACE SCIENCE MISSIONS

Mark D. Johnston[(1)], **Mark Giuliano**[(2)]

[(1)]*Jet Propulsion Laboratory, California Institute of Technology,*
4800 Oak Grove Drive, Pasadena CA 91109 USA, mark.d.johnston @ jpl.nasa.gov
[(2)]*Space Telescope Science Institute,*
3700 San Martin Drive, Baltimore MD 21218 USA, giuliano @ stsci.edu

ABSTRACT

We have developed an architecture called **MUSE** (Multi-User Scheduling Environment) to enable the integration of multi-objective evolutionary algorithms with existing domain planning and scheduling tools. Our approach is intended to make it possible to re-use existing software, while obtaining the advantages of an explicitly multi-objective optimization algorithm. Among other considerations, it enables multiple participants to actively engage in the optimization process, each representing one or more objectives in the optimization problem. As an initial application, we have applied our approach to scheduling the James Webb Space Telescope, where three objectives were modeled: minimizing wasted time, minimizing the number of observations that miss their last planning opportunity in a year, and minimizing the (vector) build up of angular momentum that would necessitate the use of mission critical propellant to dump the momentum. As a second application area, we are modeling aspects of the Cassini science planning process, including the trade-off between collecting data (subject to onboard recorder capacity) and transmitting saved data to Earth. In this paper we describe our overall architecture and our adaptations for the JWST and Cassini domains. We also describe our plans for applying this approach to other science mission planning and scheduling problems in the future.

1. INTRODUCTION

Multi-objective scheduling is an approach to optimized scheduling that offers a number of advantages over the more conventional single-objective approach [1,2]. By keeping objectives separate instead of combined, more information is explicitly available to the end user or to the scheduling software system for comprehending and deciding on trade-offs among competing objectives. Multi-objective algorithms produce a set of solutions, called a Pareto surface (aka trade-off space), where no solution is strictly dominated by another solution for all objectives. Particularly when objectives cannot be cast onto commensurate scales, visibility into the Pareto trade-off space can be extremely valuable. Algorithms for solving multi-objective problems have been developed that are effective in building up populations of candidate schedules that trace out an approximate Pareto frontier with reasonably uniform sampling. However, adapting a multi-objective scheduling approach to an operational setting is faced with at least two significant additional challenges:

- the often high dimensionality of the objective space can be difficult to convey to users using conventional graphical user interfaces: this makes it difficult to see overall patterns and trade-offs, or to see the effects of limiting objective or constraint value ranges
- the nature of many multi-objective scheduling problems requires multiple users to be heavily involved, each such user contributing one or more objectives that reflect their interest in the outcome of the scheduling process: thus there is a tightly integrated multi-user aspect that must be considered

We are applying a multi-objective scheduling approach to two major space science missions that amply exemplify these challenges: the James Webb Space Telescope (JWST), scheduled for launch in 2013, and the Cassini mission to Saturn, currently in its first extended mission. In this paper, we describe the nature of some of the user interface challenges that these kinds of missions present, and the techniques we are investigating to overcome them.

2. APPROACH

We have developed an architecture called MUSE (Multi-User Scheduling Environment) to integrate pre-existing scheduling components (e.g. scheduling engines and user interfaces) into a multi-objective scheduling framework. The MUSE architecture integrates both generic and application-specific components. Among the generic components is a means for visualizing objective value spaces for schedule populations, for registering objective limits and acceptable ranges, and for collaborative convergence on mutually acceptable schedules for multiple users. Our

Proc. 'IJCAI–09 Workshop on Artificial Intelligence in Space', Pasadena, California, US 17–18 July 2009 (ESA SP-673, September 2009)

approach to visualization includes a variety of techniques to meet the challenges noted above of higher-dimensional objective spaces, including 2- and 3-D projections of the Pareto frontier, histograms and other depictions of values in different dimensions, and attribute exploration techniques that have been successfully used in a number of data visualization applications. We have adapted elements common to mixed-initiative user interfaces that can be applied to our domain. The overall architecture is described in Section 3, and its application to two representative space science missions in Section 4. We summarize our conclusions in Section 5.

3. ARCHITECTURE

The MUSE architecture is illustrated in Figure 1. Several drivers have led to design decisions as they relate to the architecture:

- MUSE is intended to *integrate* with existing tools as easily as possible, to leverage existing work in many domains
- The collaborative elements of MUSE require *persistent storage* of various types of schedule data, hence a server-centric architecture
- Both *online and offline collaboration* need to be supported, in consideration of users working across multiple time zones — thus live interaction is available but not required

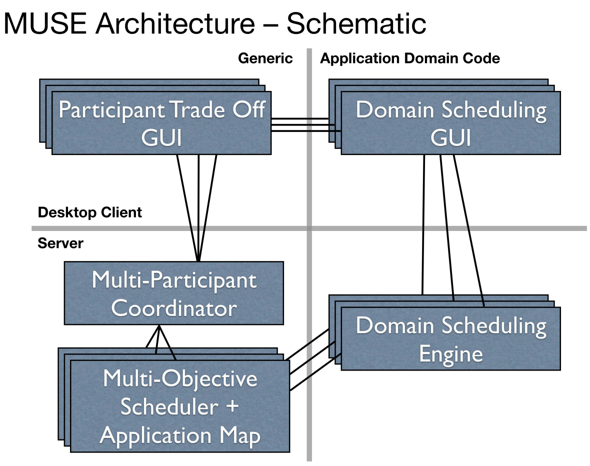

Fig. 1: Architectural overview of MUSE – see text for description

We distinguish server components (Fig. 1 lower half) from those resident on the user's workstation. We also distinguish generic components (left) from those that are highly domain specific (right). The architecture is designed so that domain specific components can be run as separate processes or can be compiled into the same image as the generic code.

We have adopted the familiar threaded email or newsgroup interaction model as a metaphor for how MUSE interacts with individual participants. Such interaction can be either on- or offline, in that one can tell upon returning to the interface what has changed since one was last present. This is important in settings where participants may use the system in an infrequent episodic manner.

On the server side, the Multi-Participant Coordinator acts as a central "clearing house" for schedule data, participant's selections, and scheduling runs. It provides a REST-based web application interface that communicates with the individual participants, providing up to date schedules, schedule status, and other participants selections of objective value ranges. The Multi-Objective Scheduler is an implementation of an evolutionary algorithm [1,2] to evolve a population of candidate schedules towards the Pareto-optimal surface. While various algorithms could be employed here, we are presently using a variant called Generalized Differential Evolution 3 [3,4]. More details about this algorithm and how it performs on some relevant domains may be found in [5]. The Application Map provides a transformation between decision variable values and domain-specific scheduling decisions as represented and evaluated in the Domain Scheduling Engine components. The Multi-Objective Scheduler supports parallel evaluations of schedules, which can frequently help speed the generation of a Pareto surface for participants.

The Domain Scheduling Engine is the application-specific scheduling software that MUSE uses to evaluate candidate schedules. This evaluation utilizes the decision variable values, and can potentially perform internal conflict resolution or optimization steps on its own before returning a set of objective function values to the Multi-Objective Scheduler. These values are used by the evolutionary algorithm to evolve the candidate population towards a well-sampled Pareto surface.

Just as Domain Scheduling Engines can be highly application specific, so are Domain Scheduling GUIs. These GUIs often already exist in many domains and are able to display and manipulate aspects of the scheduling problem that are not common from one domain to another. MUSE is intended to integrate with such GUIs, e.g. to invoke the GUI on one user-selected schedule for detailed examination and assessment.

A key function of the Participate Trade-Off GUI is visualization of the objective space of the problem, in order to comprehend trade-offs and develop a solution acceptable to all participants. For 2- and 3-dimensional objective spaces, there exist commonly used techniques for visualization that can convey the selection possibilities of the candidate schedule population. However, as the dimensionality of the objective space increases, this becomes more and more challenging [6,7]. We are investigating a number of techniques in this context for displaying higher dimension objective spaces, including:

- parallel coordinate plots
- "brushed" histograms or scatter plots that indicate correlations among attributes
- display of neighbours of selected points when projected to 1- or 2-D displays
- use of multi-touch displays for rapid and intuitive manipulations of selections and views

We expect that user preferences will play a crucial role in this area, and that a wide range of visualization options should be provided to accommodate the wide range of user preferences. We anticipate defining a "plug-in" mechanism so that it is easy to add additional visualization strategies as they become available.

A sample screen from a prototype Participant Trade-Off GUI is shown in Fig. 2, in this case for the 3-objective JWST domain (described below). With the Participant Trade-Off GUI users can view a set of candidate schedules, select limit ranges on objective values, and see what other users have selected. They can examine trade-off opportunities objective by objective and update their selections, and see the overall intersection of acceptable ranges from *all* participants. The ultimate goal is the convergence of all participants to a single selected baseline schedule; should this not occur, MUSE does not preclude any specific process from arbitrating differences and making a final selection.

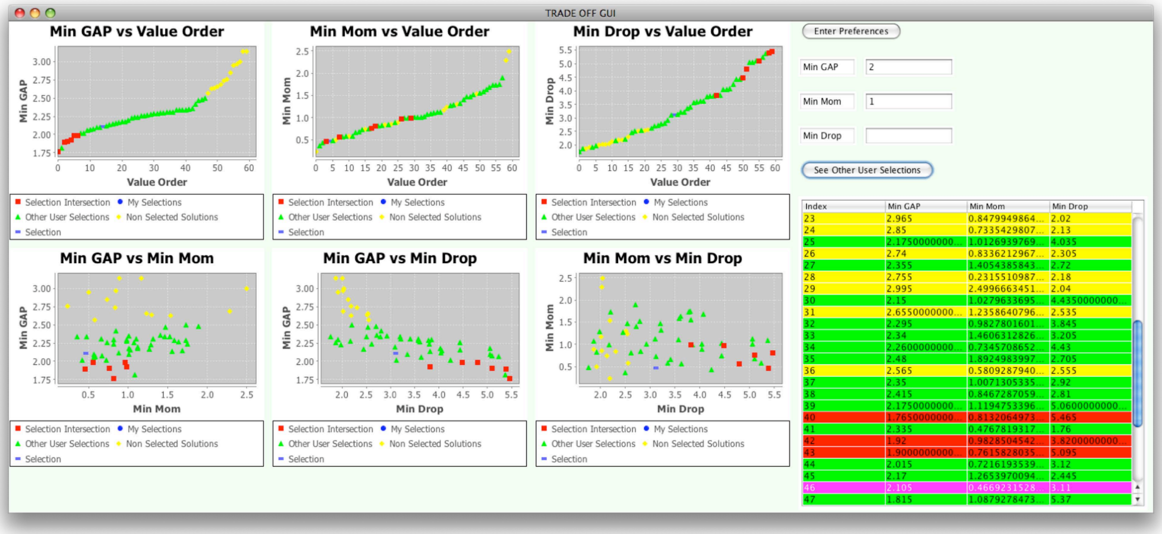

Fig. 2: a prototype of the Participant Trade-Off GUI for a 3-objective domain (JWST).

4. APPLICATIONS

We have applied the architecture described above to two very different space mission applications, which we briefly describe in the following subsections.

4.1. James Webb Space Telescope

James Webb Space Telescope (JWST, Figure 3) will be the premier astronomical facility of the next decade, replacing two of the current Great Observatories, Hubble Space Telescope (HST) and Spitzer Space Telescope (SST) as a uniquely capable space-based observatory with highly ambitious scientific objectives. Scheduled for launch in 2014, JWST will have a 6.5m primary mirror diameter (compared to 0.85m for SST, and 2.4m for HST), and will primarily observe in the infrared (like SST, and in contrast to HST's primarily optical and UV sensitivity).

Scheduling a mission such as JWST requires the balancing of many factors [8]. Clearly, such an expensive and unique facility must be utilized as efficiently as possible, and minimizing any wasted time is a primary objective. At the same time, the lifetime of the observatory is limited by consumables such as propellant for reducing momentum build-up in the spacecraft's reaction wheels. Thus, optimization of the JWST schedule is determined by multiple simultaneous objectives, for which there is no well-defined trade-off mechanism that would permit definition of a single combined objective. Multi-objective techniques that keep the objectives separate permit explicit visibility and management of the multiple trade-offs that are necessary to generate a balanced overall schedule for JWST.

For JWST, two of the primary objectives are minimizing schedule gaps, and minimizing the number of late observations, i.e. that miss their last scheduling opportunity. The more unusual objective is that of reducing angular momentum build-up in the spacecraft reaction wheels, caused by a complex interaction of pointing direction, roll angle, and solar radiation pressure on the tennis court-sized sunshade. Angular momentum build-up must be compensated by firing the spacecraft thrusters, which consumes propellant and thus is potentially a limiting factor on mission lifetime. The angular momentum resource constraint has several important features: it is a 3-dimensional vector additive quantity that applies both as a hard constraint and as a preference. The contribution to angular momentum build-up of any particular observation is a function of when it is scheduled and of the roll angle at which it is scheduled.

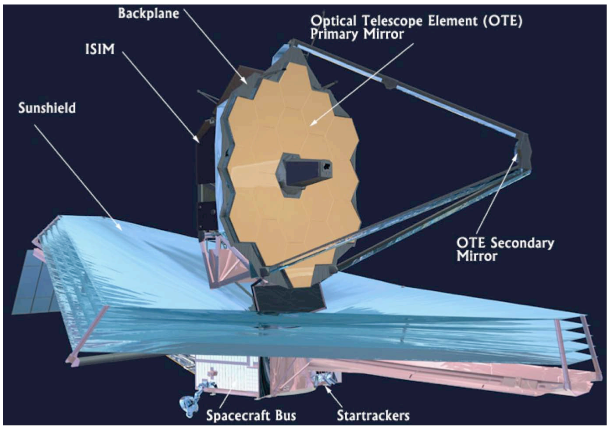

Fig. 3: Illustration of James Webb Space Telescope illustrating the segmented primary mirror and the very large sunshade.

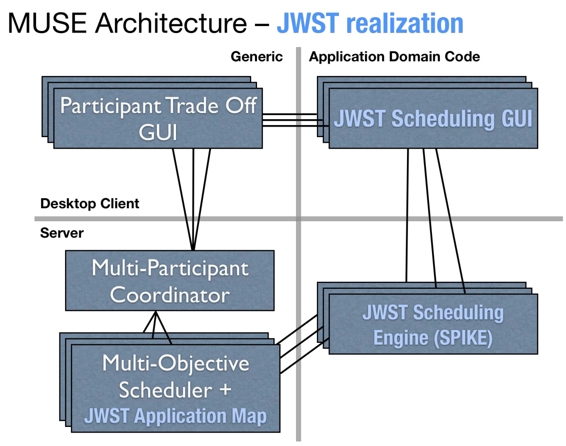

Fig. 4: Adaptation of the generic MUSE architecture for the JWST scheduling domain.

The adaptation of the generic MUSE architecture to JWST is illustrated in Fig. 4. As the JWST domain scheduler we used Spike [9], implemented in Lisp. The MUSE infrastructure is implemented in Java with the JavaFX scripting language providing user interface functionality. The two systems are integrated via a client-server socket interface that can be readily supported on both sides of the interface. This allows for the exchange of decision variable values from the multi-objective optimizer, and the receipt of objective values in return. Results from the application of the multi-objective optimizer in this manner have been reported elsewhere [10,11].

Fig. 2 illustrates the Participant Trade-Off GUI operating in the JWST context, show a display of the three objectives described above. This particular visualization shows a rank ordered plot of each objective value in the top three graphs, and the three 2-D projections in the bottom three. All of the points are cross-linked, in that selection of any point in any of the graphs, or any row of the table, will highlight the selected point on all of the other graphical and tabular views. The selection of an objective value range (via the entry boxes, upper right) highlights the selected subpopulation. In addition, the user can view *other* participants selections, and the overall intersection of all objective ranges. Finally, the user can publish their own selections to be available to other participants.

Fig. 5: The Cassini/Huygens spacecraft. The white-suited figure at lower left shows the scale.

4.2. Cassini

As a second application area, we are modeling several aspects of the Cassini science planning process [12], including the trade-off between collecting data (subject to onboard recorder capacity) and transmitting saved data to Earth, which requires a maneuver to point the high-gain antenna to Earth. The choice of downlink timing and ground-based antenna size (70m vs 34m) has a major impact on how much data can be collected and transmitted, and propagates back to the different science teams in terms of which instruments are in use and in which modes. Thus, there is a natural framing as a multi-objective optimization problem.

The Cassini spacecraft (Fig. 5) was launched in 1997 and since 2004 has been in orbit around Saturn. Cassini is a 3-axis stabilized spacecraft with 12 diverse science investigations, including 6 for optical and microwave remote sensing, and 6 for fields/particles/waves. The mission has been a spectacular success, with 260 scientists from 17 countries participating in the scientific data analysis and follow-up. The spacecraft communicates to Earth primarily through a high-gain antenna that must be pointed at Earth to use, sending back of order one Gigabyte of science data per day. During these downlink periods, most of the pointed instruments cannot be used. Thus the timing of science observations and of the downlinks must be scheduled very carefully with respect to interesting observing opportunities, in order to collect and return as much science data as possible while not overfilling the onboard recorder.

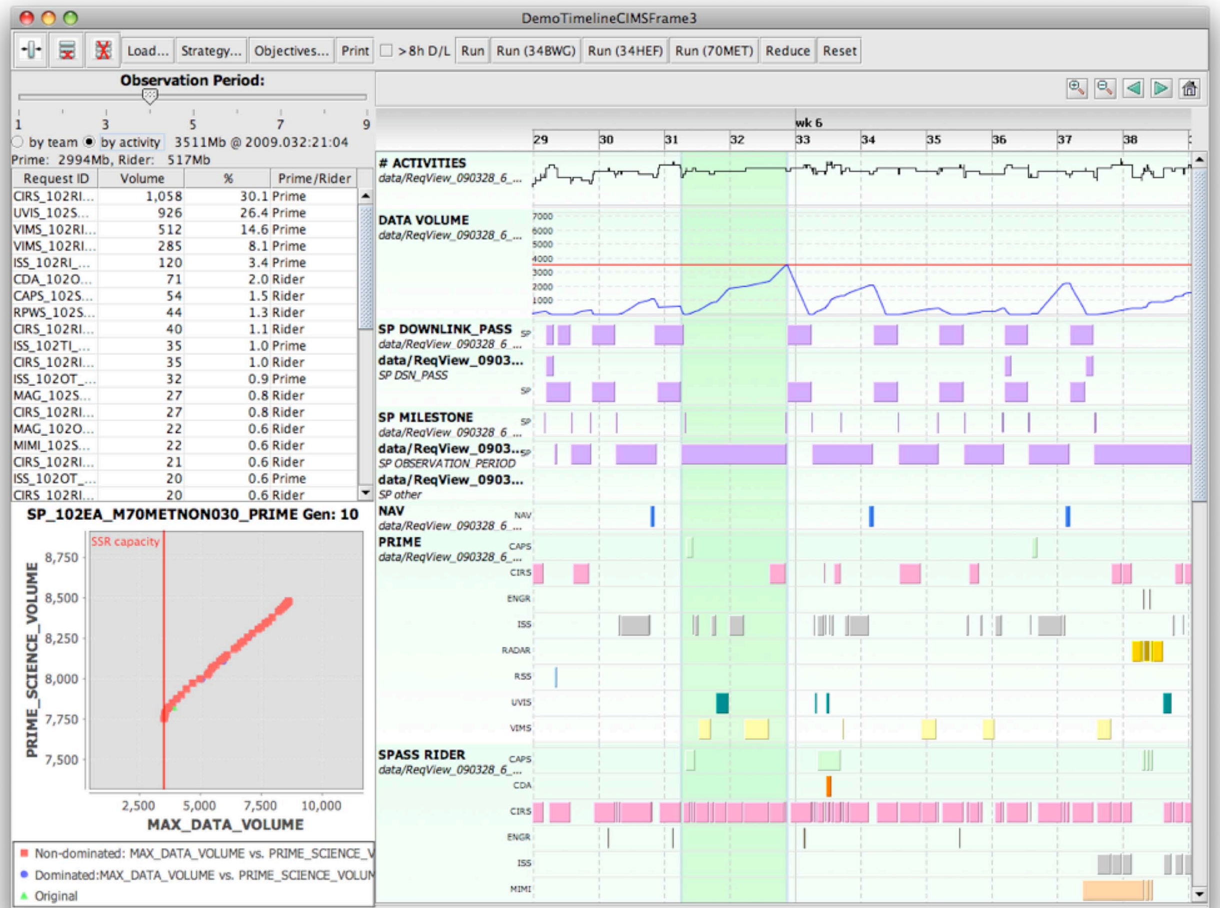

Fig. 6: A view of a Cassini schedule illustrating the Pareto trade-off in the lower left and a Gantt chart view of the various scheduled activities. Onboard data storage is limited to the value indicated by the red line in the Gantt view (second chart from top).

One of the Cassini objectives that we have modeled is based on this onboard recorder capacity limit. While this could be modeled as a *constraint* that must not be violated, we have chosen instead to define an *objective* to minimize the maximum data volume recorded, accounting both for the collection of data by the science instruments, and the dumping of data to the ground. Thus the schedule can be in an infeasible state while it is being worked on, which is useful since the degree of violation of the constraint is very visible to the user. As a second objective, we have chosen to maximize the total science data volume collected. The initial set of activities to be scheduled is defined by the science teams working with the science planners. The strategies that can be employed for improving the schedule with respect to data volume include:

- Extending or reducing the planned downlink opportunity windows, with a corresponding decrease or increase in the time spent collecting science data
- Changing a 70m contact to a 34m one or vice versa: a 70m contact can download nearly three times as much data, but can be more difficult to obtain.
- Performing an across-the-board reduction in data collected, achievable in an instrument-dependent way (e.g. possibly by switching to a less data intensive operational mode).

These strategies are encoded in the decision variables passed to the scheduling engine.

Fig. 6 shows a domain-dependent GUI illustrating this problem for a 10-day schedule period, illustrating a 2-D Pareto surface derived from the multi-objective evolutionary algorithm. As the user selects points on the Pareto frontier, the Gantt view changes to show the detailed implementation of that schedule. The tabular view on the right shows all of the contributors to the recorded data volume at the start of each downlink window. This table includes both primary and secondary ("rider") activities, and can be sorted by data volume, science team, or activity identifier.

5. CONCLUSIONS

We have described the MUSE Multi-User Scheduling Environment as an architecture for multi-user multi-objective scheduling. This problem is common to many space science missions and scientific facilities. To elaborate the necessary features and implementation trade-offs, we have adapted this architecture to two different domains: JWST scheduling, and Cassini science planning. While these adaptations are by no means complete, they have shown the significant promise of our approach, and generated interest on the part of operations teams for these missions as of potential assistance.

Future plans include the adaptation of MUSE to additional missions to both validate our overall approach, and to provide a framework for broader use. We are also actively exploring other visualization approaches that can be used for higher dimension objective spaces. The combination of improved schedule comprehension and visibility, along with collaborative schedule development, offers the potential for a significant advance in scheduling support for future missions.

ACKNOWLEDGEMENTS

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

REFERENCES

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

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

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

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

5. Johnston, M. D. (2006). "*Multi-Objective Scheduling for NASA's Deep Space Network Array*." In IWPSS-06. Baltimore, MD, Space Telescope Science Institute.

6. Spence, Robert (2001), *Information Visualization*, ACM Press, Addison-Wesley.

7. Tidwell, Jenifer (2005), *Designing Interfaces*, O'Reilly.

8. 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.

9. 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.

10. Giuliano, M., Rager, R., Ferdous, N. (2007) *Towards a Heuristic for Scheduling the James Webb Space Telescope*. In ICAPS. Providence, Rhode Island. 160-167.

11. Giuliano, M,, and Johnston M. D., (2008). "*Multi-Objective Evolutionary Algorithms for Scheduling the James Webb Space Telescope*", In ICAPS, Sydney Australia. 107-115.

12. Paczkowski, B. G., and Ray, T. L. (2004), *Cassini Science Planning Process*, in SpaceOps 2004