Visualization Tools for Multi-Objective Scheduling Algorithms

Mark Johnston
California Institute of Technology

Visualization Tools for Multi-Objective Scheduling Algorithms

Mark E. Giuliano and Mark D. Johnston

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

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109
mark.d.johnston@jpl.nasa.gov

Extended Abstract

Multi-objective algorithms for scheduling offer many advantages over the more conventional single objective approach. By keeping user objectives separate instead of combined, more information is available to the end user to make trade-offs between competing objectives. Unlike single objective algorithms, which produce a single solution, multi-objective algorithms produce a set of solutions, called a Pareto surface, where no solution is strictly dominated by another solution for all objectives. Algorithms for solving multi-objective scheduling problems have been developed that are effective in building a uniformly sampled approximation of the Pareto surface. The goal of this demonstration is to present tools that allow end users to explore the Pareto surface trade-off space in order to select a single solution for execution. This is challenging in at least two manners: First, the objective trade off space often has a high dimensionality making it hard for users to see patterns in the data using conventional graphical interfaces; Second, 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. We present features of the Multi-User Scheduling Environment (MUSE) that provides the ability to visualize higher dimension objective value spaces, and for multiple users to converge on mutually acceptable schedules.

System Architecture

The visualization tools described below are part of the Multi-User Scheduling Environment (MUSE) overall architecture, as illustrated in Figure 1 (Johnston and Giuliano 2009, Johnston and Giuliano 2010). MUSE provides a generic environment for integrating existing tools (where they exist), providing persistent storage for various types of schedule data, and supporting both online and offline collaboration (in consideration of distributed users working across multiple time zones). MUSE incorporates server components (Fig. 1 lower half) as well as components that are resident on the user's workstation. MUSE distinguishes generic components (left) from those that may be 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.

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 an interface that communicates with the individual participants, providing up to date schedules, schedule status, and other participant selections of objective value ranges. The Multi-Objective Scheduler provides the evolutionary algorithm optimizer that evolves a population of candidate schedules towards the Pareto-optimal surface. 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.

Visualization Tools

The example visualizations shown in this demonstration will be based on an application of MUSE to scheduling the James Webb Space Telescope (Giuliano and Johnston 2008). In this domain there are three objectives. To minimize gaps in the schedule, to minimize momentum build-up in telescope reaction wheels used to move the telescope, and to minimize observations which would be dropped as they missed their last opportunity to schedule.

The main focus of this demonstration is the Participant Trade Off GUI. The goal of this tool is to provide users of the system the ability to explore a Pareto-surface of potential solutions and to converge on acceptable solutions for execution. A challenge for the MUSE system is visualizing Pareto-surfaces. The traditional approach is to display the surface as a series of X-Y trade-off plots. For example, Figure 2 illustrates a trade off surface for a JWST schedule run. Although, X-Y plots are intuitive to understand they have several problems. First, the number of plots grows geometrically as the number of objectives increases. Second, it is hard to connect a point in one plot with the corresponding points in other plots. An alternate view of the data is provided in Figure 3 by a parallel coordinate plot that solves the problems with X-Y plots. In this plot each solution is represented by a single colour coded line. The values are plotted horizontally on a normalized scale. Coordinate plots solve the problems with X-Y plots but can be unreadable with a large number of points on a Pareto-surface. The important point here is that no single view of the data is always best and that the interface needs to provide multiple views. With this goal the MUSE interface provides several additional features that allow the user to dynamically explore a Pareto-surface. First, MUSE provides a tabular view of the data that supports dynamic sorting (figure 5 top). Second, we provide a plot for each criteria that graphs the criteria values in order (Figure 5 bottom). Third, the user can select a solution in one plot and have the point highlighted in all of the plots (Figure 5). Each of the plots is linked so selection a solution or region in one plot highlights the corresponding solution or region in other plots. Ongoing MUSE development is exploring additional dynamic graphical capabilities such as the ability to collapse a N dimensional objective space to an N-1,2,... dimensional objective space.

Multi-Objective applications often have different constituents that are more or less concerned with specific mission objectives and criteria. For example, engineering staff may be more concerned with telescope lifetime issues such as momentum usage in JWST. In contrast science teams would be more concerned with not dropping observations (i.e. missing the last opportunity to schedule an observation). A goal of the MUSE system is to allow multiple constituents to converge on an acceptable region of the Pareto-surface. To this end the MUSE system models different users for an application (e.g. JWST engineering, JWST science operations). Users can login and select an active schedule interval to work on from a list of active intervals. For a schedule interval a user can select preferred regions of the solution space. Figure 5 shows the interface after the JWST engineering staff has entered a preference for a momentum build up. User preferences are stored on disk and can be viewed by other users. Figure 6 shows the interface run from the perspective of the JWST science operations staff after entering a preference for dropped observations. The interface shows solutions acceptable by both, one, and no users. The interface supports an administrative user who can make the final selection or override user preference if no agreement is available.

Conclusions and Future Work

Visualization tools for the MUSE system were described that allow multiple users to explore Pareto-surfaces. The tools provide multiple views of the data that can be configured to allow users to dynamically explore the search space. The tools allow multiple users with different priorities to specify preferences and to converge on a set of solutions acceptable to all parties. Future work on the system will add additional dynamic visualization features and will evaluate its use on domains with a higher number of objectives.

Acknowledgements

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

Giuliano, M.E., and Johnston, M.D. 2008. Multi-Objective Evolutionary Algorithms for Scheduling the James Webb Space Telescope. In *International Conference on Automated Planning and Scheduling (ICAPS)*, Sydney, Australia.

Johnston, M.D. and Giuliano, M.E, MUSE: The Multi-User Scheduling Environment for Multi-Objective Scheduling of Space Science Missions. In *IJCAI Workshop on Space Applications of AI*, Pasadena, CA.

JOHNSTON, M.D. AND GIULIANO, M. 2010. Multi-User Multi-Objective Scheduling for Space Science Missions. In *SpaceOps 2010*, Huntsville, AL.

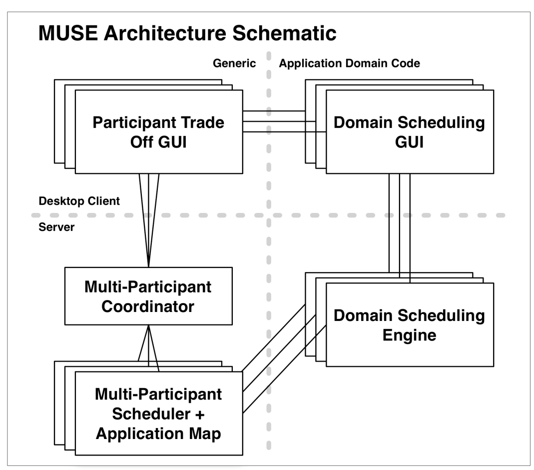

Figure 1: Multiple User Scheduling Environment (MUSE) system architecture

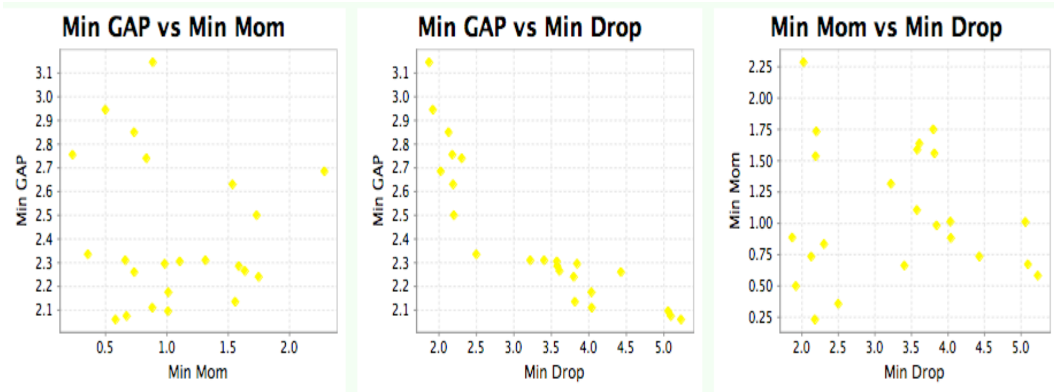

Figure 2: MUSE X-Y plots showing criteria trade-offs

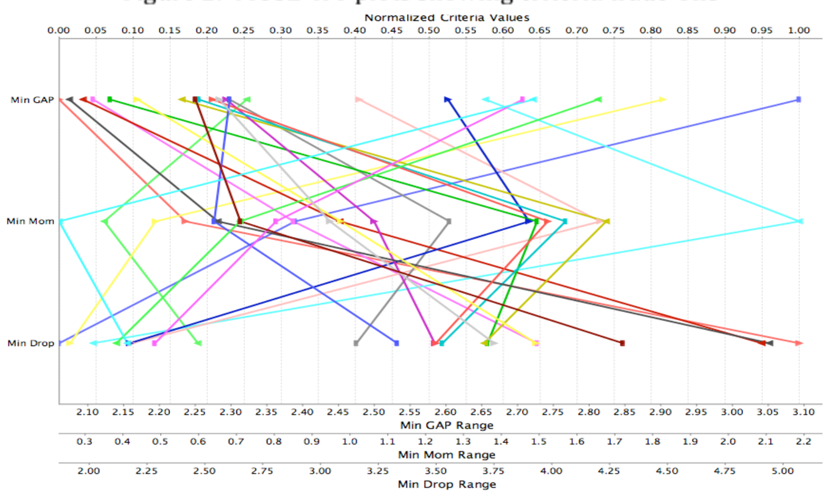

Figure 3: MUSE parallel coordinate plot

Index	Min GAP	Min Mom	Min Drop
0	2.06	0.5820426183...	5.23
1	3.145	0.8862978897...	1.87
2	2.335	0.3567022812...	2.5
3	2.945	0.4990168659...	1.92
4	2.11	0.8820475086...	4.04
5	2.685	2.2850866883...	2.025
6	2.5	1.7325837896...	2.2
7	2.31	1.3148033978...	3.22
8	2.095	1.0102924033...	5.0600000000...
9	2.63	1.5354204210...	2.19
10	2.1350000000...	1.5577325599...	3.815
11	2.24	1.7493877104...	3.8000000000...
12	2.305	1.1054975821...	3.575
13	2.265	1.6368651392...	3.61
14	2.075	0.6720651372...	5.095
15	2.285	1.5873128994...	3.58
16	2.31	0.6619029822...	3.4050000000...
17	2.85	0.7335429807...	2.13
18	2.1750000000...	1.0126939769...	4.035
19	2.74	0.8336212967...	2.305
20	2.755	0.2315510987...	2.18
21	2.295	0.9827801601...	3.845
22	2.2600000000...	0.7345708652...	4.43

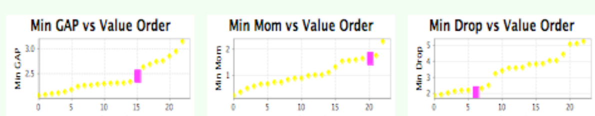

Figure 4: MUSE tabular and index plots

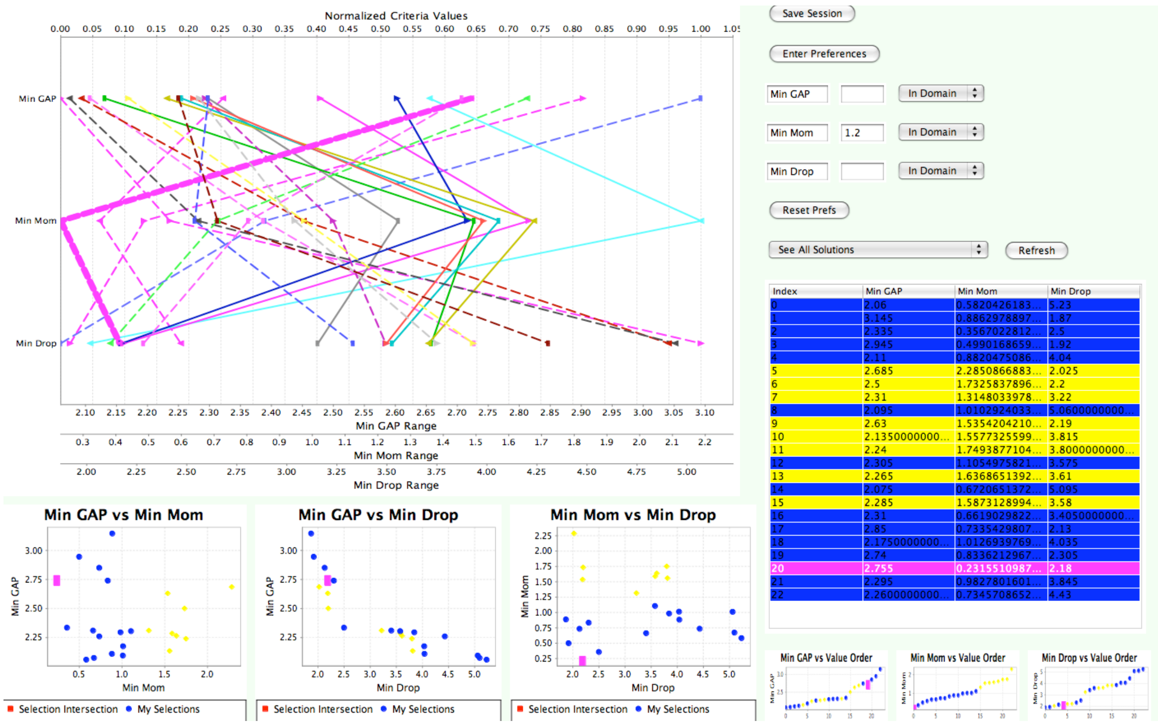

Figure 5: MUSE interface showing user selections

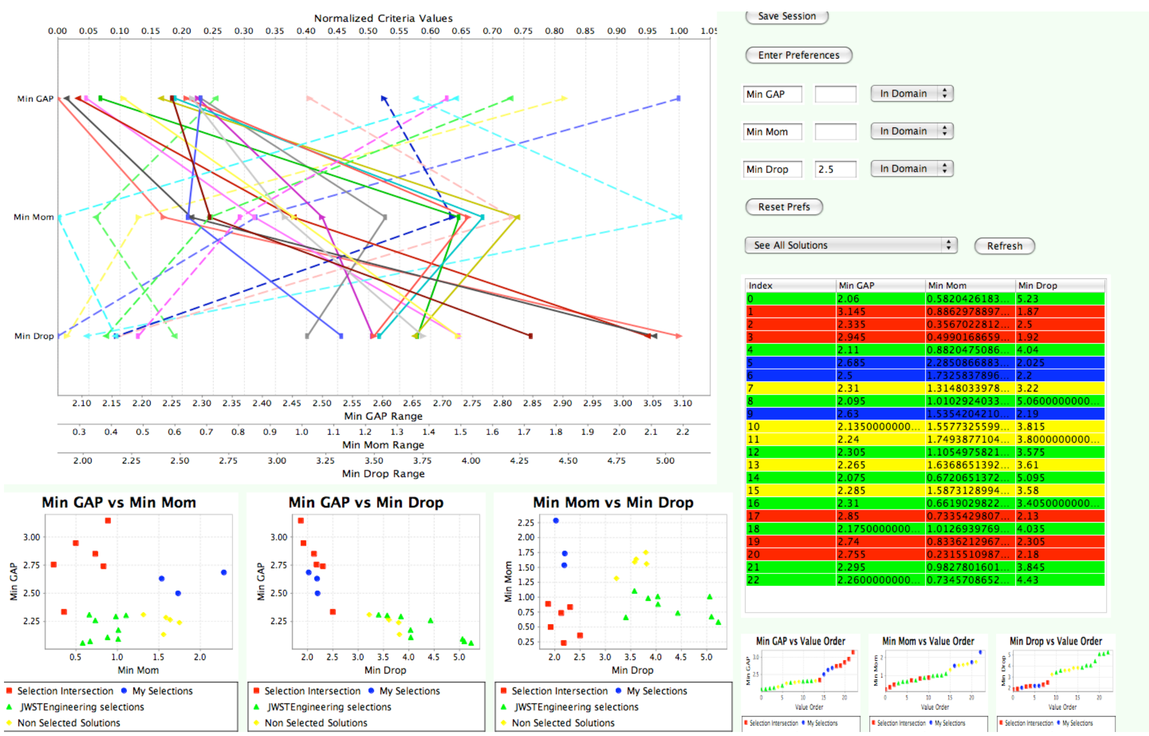

Figure 6: MUSE interface showing interactions between multiple user selections