Technologies for Mixed-Initiative Plan Management for Human Space Flight

Melissa Baltrusaitis, Karen M. Feigh, and Martijn IJtsma
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332
{mbaltrusaitis3, karen.feigh, mijtsma3}@gatech.edu

Amy Pritchett
Department of Aerospace Engineering
Pennsylvania State University, State College, PA 16801
apritchett@psu.edu

William Lassiter and Martin Savelsbergh
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta, GA 30332
wlassiter@gatech.edu, martin.savelsbergh@isye.gatech.edu

Abstract

As humans endeavor to explore Mars and other celestial bodies further afield, we are faced with a bevy of challenges unique to deep space travel. Given that astronauts have traditionally relied on ground-based mission control to produce, manage, and adjust daily flight plans as needed, one such challenge will be the time lag in communications with mission control as a crew moves further away from the Earth. This will necessitate (automated) planning systems that will provide crews greater autonomy in managing and adapting plans to reflect the current state of the mission. This paper details the progress our research team has made in developing a mixed-initiative plan management system for use on future missions to Mars and beyond. We describe the system's design and intended capabilities and provide the results of some preliminary testing with small sample plans.

Introduction

The idea of mixed-initiative planning (MIP) (Veloso, Mulvehill, and Cox 1997; Ai-Chang et al. 2004) is to "mix" the capabilities of both automatic and human planners to generate plans that are at once compliant with a potentially large number of constraints, sensible and practical in their execution, and near-optimal with respect to one or more mission objectives. Thus a successful MIP system rests on three major building blocks: (1) a formal representation of the work encompassed in the plan (hierarchically-defined activities and resources, and the constraints put upon them); (2) the interface for a human planner to reason about (and build and modify) a plan both at the detailed level of resource scheduling and the high level abstractions, and in both nominal and off-nominal conditions; and (3) computational methods to create and, ultimately, optimize plans which can potentially scale from focused, near-term off-nominal disturbance response to more strategic planning of larger sets of activities and longer durations.

It is with these three building blocks in mind that we have developed a multi-disciplinary, integrated approach to MIP that is not tied to any one method or tool. Specifically, we are pursuing a tight integration between the needs of the human planner (as identified by specialists in cognitive engineering) and of streamlined computation (as identified by specialists in large-scale optimization) by developing a system fully capable of automated planning and re-planning but still largely reliant on human feedback and direction for plan finalization. As such, plan modifications may be triggered both automatically (according to some plan monitoring protocol) and manually (according to the needs of human planners or agents), with automatic modifications requiring approval from a human agent or planner. In order to fully accommodate both of these modification schemes, our system is comprised of three distinct components, all described briefly below, working in concert with one another, with each component intended to function as one of the three aforementioned building blocks.

Work Models that Compute (WMC): A work modeling and simulation framework that has previously been used to analyze and synthesize function allocation between air traffic controllers and an automated air traffic control system, (Pritchett, Bhattacharyya, and IJtsma 2016), a pilot and an autoflight system in the flight deck of an aircraft (Pritchett, Kim, and Feigh 2014b; 2014a), and human and robotic agents in manned space flight operations (IJtsma et al. 2017a; 2017b).

Marvin: A plan display and interface tool initially built as a timeline-tracking tool for extra-vehicular activities (EVA). It is designed to allow a mission crew to interact with a plan via both direct manipulation of individual activities and higher-level manipulation of plan priorities and constraints (Miller, Pittman, and Feigh 2017).

Optimizer: A generic term for a set of plan optimization heuristics that we will be using to modify or generate plans with the goal of reaching near-optimality with respect to one or more plan objectives. The techniques that we are currently exploring are inspired by meta-heuristic and local search concepts commonly used by operations researchers to tackle (machine) scheduling problems.

Figure 1 below is intended to represent how these three components will interact with one another in the larger system. WMC and the optimizer will be tightly coupled, communicating back and forth frequently and providing an updated plan to Marvin to allow the crew to review and finalize the modified plan.

Each component will be discussed in more detail in subsequent sections, but we first provide a brief synopsis of prior related work and a description of the plan modeling frame-

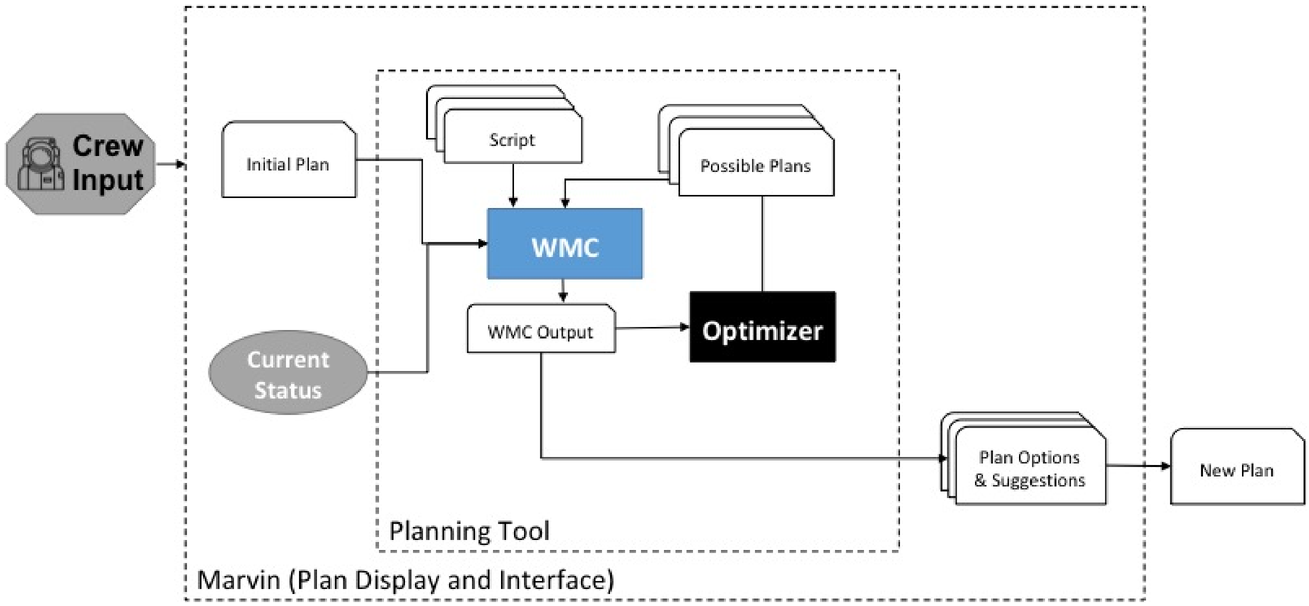

Figure 1: A visualization of our system architecture

work that we have developed for the larger system.

Related Work

Various technologies for planning have been defined in the literature. Of note, mixed-initiative planners have been studied and applied in a range of domains, including unmanned spacecraft science missions (Nothdurft et al. 2015; Biundo et al. 2011; Myers et al. 2001; Wang et al. 2013; Veloso, Mulvehill, and Cox 1997; Ai-Chang et al. 2004; Maldague et al. 1998). However, these developments have focused primarily on the technology and human aspects of MIP without examining the work of planning and the context within which it occurs. Within this limited framework, this area of the literature has debated the merits of "planning like the human" (interpreted as using some simple domain-specific heuristics which do not fully utilize the machine capabilities) (Seegebarth et al. 2012; van Wezel and Jorna 2009) or "using AI to plan" (interpreted as automatically generating plans which the human may or may not be able to interact with and modify, yet are reported to need intervention some 30-40% of the time, particularly when the system does not have full knowledge of the goals or constraints) (Ai-Chang et al. 2004; Cegarra and van Wezel 2012). In many of the "using AI to plan" cases, the underlying representation of the plan's rationale has not been clear to the human, a difficulty reflected in the costs of user interfaces associated with these various systems, reported as being up to 70% of the cost of development (Cegarra and van Wezel 2012).

Critiques of MIP systems not developed around explicit analysis of the underlying "work" to be performed have noted key obstacles to their implementation. For example, MIP has historically separated the "planning" of activities from the "scheduling" of resources; in contrast, the work of space flight planning must simultaneously consider what is feasible in terms of resource constraints when planning activities (Myers et al. 2001; Maldague et al. 1998). Likewise, human planners were found to use in their work non-exclusive decompositions reflecting activities operating upon shared resources, and to create annotations within structures intended to define resources to instead reflect more abstract information about the plan. From a cognitive engineering perspective, the obstacles faced by the current state of the art reflect a lack of emphasis on understanding the work of planning.

An awareness of the context of planning, i.e. recognizing that there can be multiple different strategies for planning based on the scenario within which planning is taking place, is equally critical to building an effective MIP system. For example, the notion of cognitive control describes how human planners may employ different strategies depending on the balance of resources (time-available, relevant knowledge, etc.) versus demands (time pressure, unfamiliar situations, etc.) (Hollnagel 1993). Feigh's work, for example, has demonstrated how different strategies for airline operations planning can be supported by different interfaces in such a way that simpler, more-efficient-but-less-optimal strategies can be easily invoked when efficiency is to be valued and a different interface can be toggled to when time allows for more strategic approaches (Feigh and Pritchett 2010).

Rather than relying on AI-based planning algorithms, we are employing methods developed in operations research (OR), a field that has long examined problems in planning and scheduling, that have been extremely successful in solving large, complex, practical optimization problems. While OR began as the study of deterministic and single-objective (and thus largely impractical) problems, focus in recent years has shifted to fields with more practical applications—

stochastic and robust optimization (Kleywegt and Shapiro 2007; Bertsimas, Brown, and Caramanis 2011), with which plans that explicitly account for quantifiable future uncertainties can be created, online optimization (Sgall 1998; Kalai and Vempala 2005), with which plans can be adjusted in real-time based on newly revealed information, and multi-objective optimization (Ehrgott 2010), with which tradeoffs between competing plan objectives may be explored and quantified. Local search techniques and heuristics have also received quite a bit of attention and have proven to be quite useful in solving large scheduling problems (Aarts and Lenstra 2003). To our knowledge, these more advanced OR techniques have never been integrated into a single decision framework akin to the one necessary for deep space exploration missions.

Plan-Modeling Framework

A robust plan-modeling framework is crucial to the efficacy of any MIP system, and especially so for our system given that it requires communication across three separate platforms. The common framework that we have developed is based on three main high-level constructs: activities, agents, and resources. All members of these three constructs have their own set of attributes that we have divided into three separate categories: characteristics, current state, and plan information. In the operation of our system, characteristics are static input data, current state is dynamic input data, and plan information is output data. A more detailed discussion of each construct and its relevant attributes is given below.

Activities

Activities comprise everything that must be done in a given time frame and are organized in a four-tier structure, with activities (highest level) subdivided into tasks, tasks into subtasks, and subtasks into procedures. This organizational structure helps to facilitate flexibility with respect to the amount of granularity required for planning or re-planning. In the context of planning, we have developed and categorized relevant attributes of activities as follows:

Characteristics Relevant activity characteristics include

- A unique identifier (e.g. activity name)
- Earliest (Latest) allowed start (completion) time
- Location and duration
- Activity tier, and parent (one tier higher) and child (one tier lower) activities, if applicable
- Any preceding or succeeding activities
- Difficulty level (subjective, rated on to a numeric scale)
- Agent skills and/or resources required for execution.

Current State The only current state information needed for an activity is a flag indicating whether or not it is already included in the current plan.

Plan Information Plan information for an activity includes the agent(s) performing the activity, its start and end times, and the identifiers of any resources it is using.

Agents

An agent is any entity capable of performing work; a plan's list of agents may include human crew members, robonauts, robotic arms, astrobees and other such entities. The attributes of agents that we have deemed relevant in the context of planning are as follows:

Characteristic Relevant agent characteristics include a unique identifier (e.g. agent name), oxygen and/or energy consumption rates, agent skills (in order to match certain agents with certain activities), and movement speed/range of motion.

Current State Current state information for agents includes location, resources in possession (e.g. tools for completing a maintenance activity), agent fatigue level (calculated according to difficulty levels of activities executed) and times available (times in the plan when the agent is not scheduled to be working).

Plan Information Plan information for a particular agent is simply the set of activities that the agent has been tasked with executing in the plan.

Resources

A resource is an inanimate object or supply which must be utilized to perform certain activities. A resource may be a tool, replacement part, EVA suit, oxygen tank, power cell, or other such implement. Relevant attributes of resources in the context of planning are as follows:

Characteristic Relevant resource characteristics include a unique identifier (e.g. resource name + number if there are more than one of something), energy consumption rate, maximum battery and/or oxygen capacity (when applicable), and storage location.

Current State Current state information for resources includes location, times available, and oxygen and battery levels when applicable.

Plan Information Plan information for a particular resource is, similar to that of an agent, simply the set of activities that utilize the resource in the plan.

These three constructs and their relevant attributes form the backbone of our planning structure and must be interpreted and utilized by each component of our system architecture. We now proceed to discussing this architecture in greater detail.

System Architecture

As was mentioned in the introduction, we have designed our MIP system to address the needs of both the human planner and streamlined computation. Of vital importance to the human planner is an adequate representation of the work to be completed and a conception of how this work will flow in a given plan. Streamlined computation requires a structure that lends itself to the creation of robust, efficient modification and optimization algorithms. With these ideas in mind, this section is dedicated to our efforts so far in (1) the modeling and simulation of work and (2) the development of plan optimization techniques.

Modeling Work

Provided the emphasis on the human in MIP, we have extended this influence into the development of our model's work constraints. By taking a human-centered approach, we have not only considered traditional technological constraints, but we have also examined cognitive, social, and physical factors that influence work. Thus, we have defined the following requirements by resource, temporal, and agent activities.

Resource Constraints

- Activity X requires Y resource <oxygen, CO2, water, power, thermal> amount
- No more than Y resources used in Z time period

Temporal Constraints

- Activity X <before, after, at same time> Activity Y
- Perform Activity X at Time T
- Perform Activity X at RelativeTime R <prior, after> to Time T
- Perform Activity X at RelativeTime R <prior, after> to Activity Y
- Combine Activity X and Y into Activity Z in this order
- Break Activity X into Activity Y and Z at Step J
- Add task X <now, Time T>
- Remove Activity X

Agent Constraints

- Assign Activity X to Agent Y
- Split Activity X across Agents Y and Z

To further examine the factors that influence work in a MIP system, specifically astronaut-system interactions, we developed three scenarios that span varying levels of safety and time considerations. These scenarios include emergency, mission impact, and optional events, with each defined in detail in Figure 2 above. Building on these scenarios by creating storyboards, information flow diagrams, and initial user environment design documents (Beyer and Holtzblatt 1998), we reinforced the proposed work model for the optimization algorithm and provided a foundation for initial interface design requirements.

Simulating Work: WMC

WMC is a computational simulation framework that has been developed over several years, originally as a means of evaluating function allocations between various automation, robotic and human agents (Pritchett, Kim, and Feigh 2014b; 2014a). The high-level model structure of the framework consists of a work model and an agent model, which are then simulated through time as agents performing the work in the work model. The work model is a representation of the activities of a plan as well as the resources that are required for the work. Each activity has attributes that define constraints such as its location, the resources that are used or consumed and the duration it has. An agent model consists of simple heuristics for executing each activity. WMC takes two inputs: a function allocation that assigns each activity to a performing agent and a plan that consists of steps with activities and scheduled times.

The simulation loads the plan and function allocation into its simulation core, and steps through this internal activity list sorted by scheduled time and calls associated agent models to perform the activities due at the current simulation time. When the agent model is called by the simulation's core to perform an activity, its internal checks account for constraints in the work and the agent, such as the location and availability of the required resources, the maximum number of activities each agent can perform at a time, and the location of each agent.

When constraints are not met, the agent model has simple heuristics to resolve a constraint violation. For example, when a resource for an activity is not available at its original scheduled time, the agent model will delay the activity until this resource becomes available (when the activity that was occupying the resource finishes). In case locations of resources or agents do not match, the simulation can account for required traversal times to fetch resources or change agent locations. These alterations to the plan are logged and fed back to the optimization algorithm for further iterations.

The framework furthermore logs several metrics during simulations that can subsequently be used as objectives in the optimization process. Examples of performance metrics are the total time to completion, the idle or down time of agents and the total time on task for each agent. Additionally, WMC logs metrics capture some of the coordination or teamwork that is required between agents to make a plan work. For example, instances of information sharing between two agents (based on their activities and the respective timing) are logged as a measure for the required communication. When resources need to be shared between agents, WMC logs the transfer of these resources as requirements for physical interaction between agents in which the resources are handed over. The traversal time associated with such resource management, or simply to move from one location to the other between consecutive activities, is also logged as a metric.

Optimization

The plan-modeling framework detailed in the previous section lends itself nicely to a scheduling theory-based approach to modification and optimization. Scheduling theory is a branch of OR that has grown out of efforts to solve variations of the classic machine scheduling problem: the problem of scheduling n "jobs" on m "machines" in such a way that some objective (e.g., the completion time of the last job) is optimized. In the context of automated planning for human space flight, activities can be thought of as "jobs" and agents as "machines". Many of the temporal and agent constraints discussed above have natural representations within the confines of a machine scheduling problem, and resource constraints can fairly easily be tracked at a high level as well.

Thus, the Optimizer briefly described earlier essentially

Scenarios	Time Allowable	Depth of Re-plan Required	Safety Impact	Type of Re-plan
Emergency: An astronaut loses suit pressure while performing an EVA	Minimal	Low Depth Initially (fix problem), but greater depth potential in the future	High-Safety	Automatic
Mission Impact: An experiment is running significantly over its allocated time	Moderate	Moderate Depth	Moderate Safety	Mixed (System asks for astronaut input)
Optional: An astronaut would like to add personal time into the plan	Substantial	Moderate Depth	Minimal Safety	Manual

Figure 2: Detailed description of the three scenarios. In this table, we consider: the amount of time that the optimizing system has to evaluate and present a new plan to the team – time allowable; the amount of the plan that will need to be evaluated and potentially modified – depth of re-plan; the introduction of new constraints that prioritize crew and structure health – safety impact; and the type of method used to trigger a re-planning event — type of re-plan.

endeavors to solve (or come very close to solving) a machine scheduling problem over given sets of activities and agents. Given that the problem of optimally scheduling jobs on two or more machines with an objective as simple as time to completion has been shown to be $\mathcal{NP}$-complete (i.e., computationally intractable for large instances) (Ullman 1975), solution techniques are generally heuristic in nature. Local (or neighborhood) search is one such solution technique that has proven effective in handling large instances, especially when embedded in a meta-heuristic framework, making it our method of choice for use in the Optimizer.

Our current local search algorithm is relatively simple. The "neighborhood" around a given schedule is defined to be all schedules that can be obtained from it by removing a single activity and reinserting it elsewhere. An objective value to optimize is specified at the outset, and at each iteration a schedule in the neighborhood is selected at random. If this new schedule has an objective value that is at least as good as the current schedule, the algorithm "jumps" from the latter to the former, and then repeats. Otherwise, the current schedule remains as is and a different schedule in the neighborhood is selected in the next iteration. This process continues either for a set number of iterations or until the schedule's objective value reaches a threshold, depending on the user's preference. The algorithm can also consider multiple objectives via added constraints that prevent jumping to a new schedule if doing so would cause one or more secondary objective values to exceed specified thresholds.

Given that as many as several thousand or more such jumps may be necessary to obtain a close-to-optimal schedule, examining resource availability, usage, and consumption in depth at each iteration has the potential to be computationally prohibitive. With this in mind, our algorithm treats resource-related constraints at a coarse, high level and relies on WMCs detailed evaluation process for resource-related feedback and suggested schedule changes. The section below discusses the integration between the Optimizer and WMC in greater depth.

Integration

The optimization algorithm and WMC are written as separate processes in C++. Both contain similar objects, albeit for different purposes: the optimization algorithm uses heuristics to reason using simple constraints and to construct new plans, and WMC uses the objects to evaluate a plan using a range of metrics. However, to assure the objects in the two processes operate consistently on the same data, the class attributes of each objects are populated from the same XML input file, as shown in Figure 3.

This XML file defines the activities of the plan, the available agents and resources, and provides the optimizer and WMC with information on the resource, temporal, and agent constraints. Any input from the astronaut through the Marvin interface will also be defined in this XML input format. The file follows the same basic format as described in the Modeling Framework section and thus contains static information on objects (the Characteristics attributes), an initial state of dynamic attributes (Current State attributes), and the original plan that is used as a starting point for the optimization (Plan Information attributes).

Then, during the optimization process, as different plans are constructed and evaluated in WMC, the dynamic attributes of objects change. To exchange this changing information - a plan change to be simulated in WMC and the corresponding metrics to be considered in the optimization algorithm - we use a C++ *shared memory object*. Compared to other methods like external log files or message passing, shared memory is considered the most computationally efficient method for interchanging data between processes. To reduce the required communication, only the dynamic attributes of objects are communicated through the shared memory, including:

- Scheduled time for each activity.
- The function allocation denoting which agent is assigned to perform each activity.
- Metrics of interest for the plan. Metrics in the initial im-

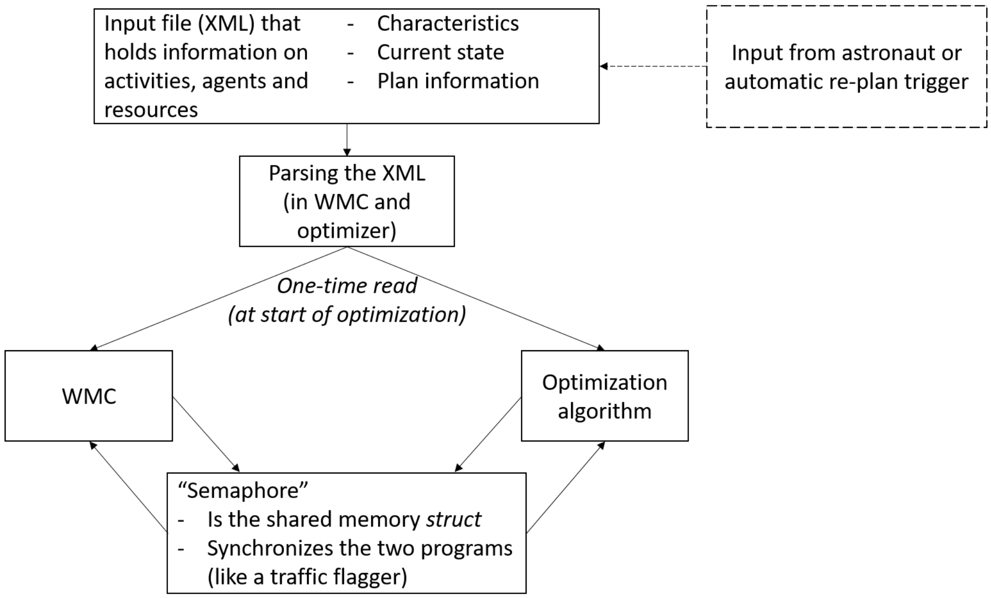

Figure 3: Diagram of the integration between the optimization algorithm and the WMC simulation framework.

plementation include the makespan (i.e., the total duration of a schedule), mental and physical workload estimates for each agent, time on task for each agent, the total number of physical resource exchanges between agents, and a list of actions that are performed past their deadline.

Sample Case Study

To validate the coupling of the optimization algorithm and WMC, we have constructed a use case emulating the replanning of one day's existing schedule to produce a feasible, modified solution. To enable this process, we have developed an example 12-hour mission schedule that reflects the work, exercise, and leisure standards that are currently employed by the Flight Planning team at NASA JSC. The names of the activities in the schedule are shown in Figure 4. Each activity in the schedule contains inherent characteristics, including location, skills, mental and physical difficulty, and temporal and resource dependencies, which are used to constrain new solutions. The activity constraints contained within the case study are designed explicitly to ensure that the schedule captures all of the plan-specific work requirements that were outlined in the Modeling Work section.

For this particular case study we have limited ourselves to considering two distinct objectives: (1) the plan's makespan (i.e. overall time to completion) and (2) workload balance among agents. The results displayed below were obtained by optimizing with respect to makespan, but the plan can be optimized with respect to workload balance as well by aggregating the difficulty levels of assigned activities across all agents and minimizing variance between agents.

Figure 5 shows the existing 12-hour mission schedule before optimization (subfigure a) and the schedule after optimization (subfigure b), in the form of time traces of when each activity is performed and by which agent. For this preliminary case study, we let the optimization algorithm perform ten thousand iterations, and subsequently simulated the resulting best option in WMC for more detailed evaluation.

The optimization process resulted in a plan with a 45-minute reduction in makespan, following a considerable amount of activity re-scheduling (e.g., the order of activities for Agent 3 has changed notably) and/or re-assignment (e.g., activity 14 is re-assigned from Agent 1 to Agent 2). All changes that involve re-assigning activities from one agent to another have been highlighted using dashed lines; most re-assignments are found between Agents 1 and 2. Agent 3 has mostly changes in the order of his/her activities. These changes result in a more condensed schedule, with a more balanced workload distribution. Finally, the makespan cannot be reduced any further since activities 38 and 39 ("Prepare Findings Report 1" and "Prepare Findings Report 2") both have constraints that require them to be executed at their respective times.

Future Work

In this paper, we present a novel approach to MIP for long-duration spaceflight that focuses on the tight coupling of three components: a work-modeling framework, an optimization algorithm, and an intuitive user interface. While the case study proved the feasibility of this design, significant work remains to present a formal solution.

Better modeling the true nature of work is one focus area. We are continuing to refine the example schedule to better reflect mission-planning constraints and manage human abilities and preferences, including reducing the spacing between activities. Additionally, we are modeling a new mission-impact scenario that is focused on the inclusion of an unplanned maintenance task, which will require direct input from ground control.

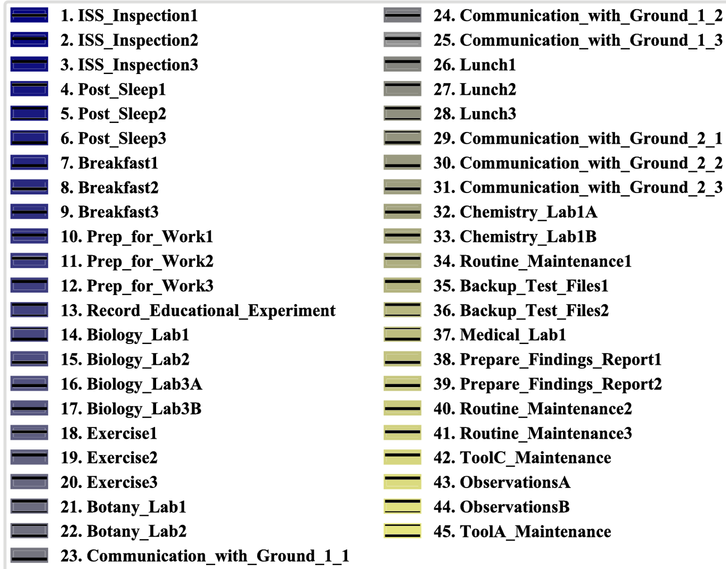

Figure 4: Activities to be completed as part of case study.

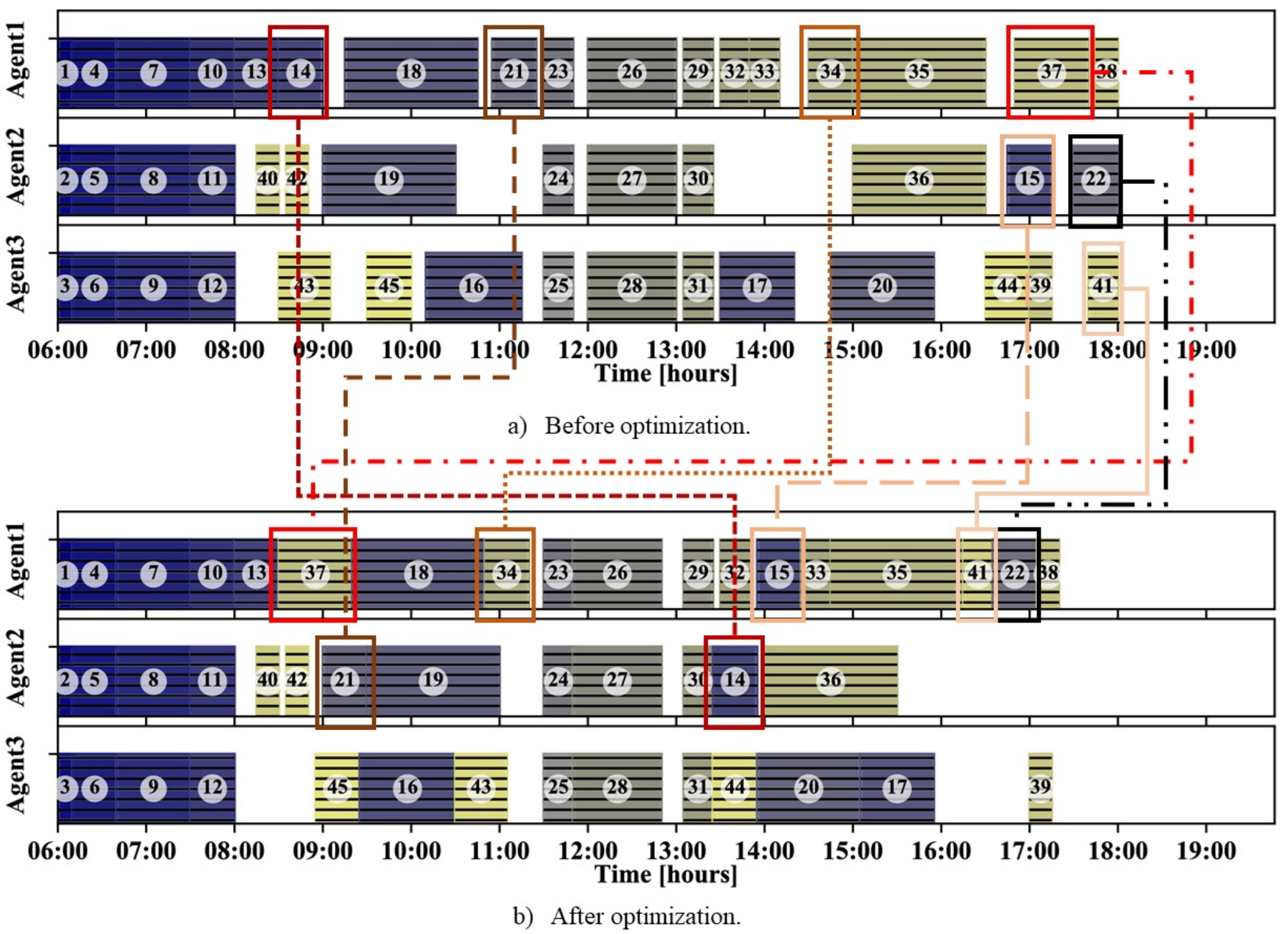

Figure 5: Time trace of existing mission schedule before and after optimization.

Future work on the WMC framework will include feeding back to the Optimizer more advanced metrics that would not be easily evaluated in the optimization algorithm, such as the required traversal of astronauts through the space station and more detailed resource-related metrics. Another possible addition to WMC is to simulate uncertainty in the action duration, allowing for estimates of the robustness of a plan to natural variation in the work.

Work will also continue on improving our local search heuristic for optimization. We anticipate developing more efficient and problem-tailored algorithms through the use of metaheuristics such as tabu search, adaptive large neighborhood search, and multi-start search, as well as implementing techniques to address new, more subtle objectives such as agent preferences, resource consumption, and total number of physical resource exchanges between agents.

We also plan to continue expanding upon the integration between the optimization algorithm and WMC, writing more metrics from WMC to the shared memory and having the optimizer use these simulation results in clever ways in follow-up iterations. One important aspect to consider in the integration is that when WMC performs its more detailed evaluation it should provide rich feedback to the optimizer, i.e., not just the metric values but also reasoning behind any delays or changes in the schedule. Likewise, when the WMC and optimizer combination is integrated with a human interface, similar kind of informative feedback should be fed back to the system's user.

Finally, the development of the interface is an intended area for future work. To date, we have developed scenarios, storyboards, information flows, and user environment designs to derive initial interface design requirements. Thus, the next step will be to convert these requirements into mockups and prototypes, and ultimately, a functional and intuitive application.

Acknowledgments

This work is funded as part of a NASA Early Stage Innovation Grant (NNX17AD12G), with Steve Hillenius and Jeremy Frank serving as Program Managers. This work is solely the responsibility of its authors and does not necessarily represent the official views of NASA.

References

Aarts, E. H. L., and Lenstra, J. K. 2003. *Local Search in Combinatorial Optimization*. Princeton University Press.

Ai-Chang, M.; Bresina, J.; Charest, L.; Chase, A.; Hsu, J. C. J.; Jonsson, A.; Kanefsky, B.; Morris, P.; Rajan, K.; Yglesias, J.; Chafin, B. G.; Dias, W. C.; and Maldague, P. F. 2004. Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems* 19(1):8–12.

Bertsimas, D.; Brown, D. B.; and Caramanis, C. 2011. Theory and applications of robust optimization. *SIAM Review* 53(3):464–501.

Beyer, H., and Holtzblatt, K. 1998. *Contextual Design*. San Diego, CA: Academic Press.

Biundo, S.; Bercher, P.; Geier, T.; Mller, F.; and Schattenberg, B. 2011. Advanced user assistance based on ai planning. *Cognitive Systems Research* 12(3):219 – 236. Special Issue on Complex Cognition.

Cegarra, J., and van Wezel, W. 2012. Revisiting decision support systems for cognitive readiness: A contribution to unstructured and complex scheduling situations. *Journal of Cognitive Engineering and Decision Making* 6(3):299–324.

Ehrgott, M. 2010. *Multicriteria optimization*. Springer.

Feigh, K., and Pritchett, A. 2010. Modeling work for cognitive work support system design in operational control centers. *Journal of Cognitive Engineering and Decision Making* 4:126.

Hollnagel, E. 1993. *Human Reliability Analysis: Context and Control*. Academic Press.

IJtsma, M.; Ma, L. M.; Pritchett, A. R.; and Feigh, K. M. 2017a. Work Dynamics of Taskwork and Teamwork in Function Allocation for Manned Spaceflight Operations. In *International Symposium on Aviation Psychology*, 554–559.

IJtsma, M.; Pritchett, A. R.; Ma, L. M.; and Feigh, K. M. 2017b. Modeling Human-Robot Interaction to Inform Function Allocation in Manned Spaceflight Operations. In *Robotics: Science and Systems, Workshop: Bridging the Gap in Space Robotics*.

Kalai, A., and Vempala, S. 2005. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences* 71(3):291 – 307. Learning Theory 2003.

Kleywegt, A. J., and Shapiro, A. 2007. *Stochastic Optimization*. Wiley-Blackwell. chapter 102, 2625–2649.

Maldague, P.; Ko, A.; Page, D.; and Starbird, T. 1998. Apgen: A multi-mission semi-automated planning tool. In *First International NASA Workshop on Planning and Scheduling*.

Miller, M. J.; Pittman, C. P.; and Feigh, K. M. 2017. Next-generation human extravehicular spaceflight operations support systems development. In *International Aeronautical Congress (IAC)*.

Myers, K.; Smith, S.; Hildum, D. W.; Jarvis, P.; and de Lacaze, R. 2001. Integrating planning and scheduling through adaptation of resource intensity estimates. In *Proceedings 5th European Conference on Planning*.

Nothdurft, F.; Behnke, G.; Bercher, P.; Biundo, S.; and Minker, W. 2015. The interplay of user-centered dialog systems and ai planning. *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Pritchett, A. R.; Bhattacharyya, R. P.; and IJtsma, M. 2016. Computational assessment of authority and responsibility in air traffic concepts of operation. *Journal of Air Transportation* 24(3):93–102.

Pritchett, A. R.; Kim, S. Y.; and Feigh, K. M. 2014a. Measuring human-automation function allocation. *Journal of Cognitive Engineering and Decision Making* 8(1):52–77.

Pritchett, A. R.; Kim, S. Y.; and Feigh, K. M. 2014b. Modeling human-automation function allocation. *Journal of Cognitive Engineering and Decision Making* 8(1):33–51.

Seegebarth, B.; Muller, F.; Schattenberg, B.; and Biundo, S. 2012. Making hybrid plans more clear to human users-a formal approach for generating sound explanations. In *Twenty-Second International Conference on Automated Planning and Scheduling*.

Sgall, J. 1998. *On-line scheduling*. Berlin, Heidelberg: Springer Berlin Heidelberg. 196–231.

Ullman, J. 1975. Np-complete scheduling problems. *Journal of Computer and System Sciences* 10(3):384 – 393.

van Wezel, W., and Jorna, R. 2009. Cognition, tasks and planning: supporting the planning of shunting operations at the netherlands railways. *Cognition, Technology & Work* 11(2):165–176.

Veloso, M. M.; Mulvehill, A. M.; and Cox, M. T. 1997. Rationale-supported mixed-initiative case-based planning. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, 1072–1077. AAAI Press.

Wang, Z.; Wang, H.-W.; Qi, C.; and Wang, J. 2013. A resource enhanced htn planning approach for emergency decision-making. *Applied Intelligence* 38(2):226–238.