ONBOARD AUTOMATED SCHEDULING FOR THE MARS 2020 ROVER

G. Rabideau[1], V. Wong[1], D. Gaines[1], J. Agrawal[1], S. Chien[1], S. Kuhn[1], E. Fosse[1], J. Biehl[1]
[1]Jet Propulsion Laboratory, California Institute of Technology,
4800 Oak Grove Drive Pasadena, CA 91109, firstname.lastname@jpl.nasa.gov

ABSTRACT

The Mars 2020 Mission, scheduled to land on Mars February 18, 2021, has developed an onboard scheduling system [1]. The rationale for the onboard scheduler is to enable the Perseverance rover to adjust its activities in response to activities taking longer or shorter than planned, or using more or less resources than expected, as effectively using these resources could significantly improve rover productivity [2]. If deployed, the onboard scheduler would be an unprecedented use of Artificial Intelligence/Autonomy onboard software in a key role for a major mission.

1 INTRODUCTION

Traditional rover planning, such as that employed by the Mars Science Laboratory, follows a long and work intensive process [2]. At any point, rover operators must balance the scientific goals of the mission with the safety and durability of the vehicle. They must consider the timing for when they will receive critical data from the rover regarding its surroundings, its pose, and the health of its hardware. All of this is directly used to inform the scope of activities the rover is allowed to perform the next day.

JPL has performed studies [2] on rover productivity under this traditional style of operation. From a mission perspective, the primary capability of a rover is being able to use its onboard instruments to measure and interact with science targets throughout the Martian landscape. As such, the “productivity” of the rover can be defined as the amount of time spent performing these science activities or driving to the targets. The study uncovered factors related to traditional operations that affected rover productivity, such as: long planning time that constrain uplink/downlink schedules, the need to insert margin into activities to guarantee their execution, an inability to take advantage of onboard knowledge, and so on.

The Mars 2020 Onboard Scheduler enables a new paradigm of operations that seeks to address at least some of these issues [1]. A ground version of the scheduler, along with its associated tools, is used by the operators to generate a plan specification in a much shorter amount of time compared to traditional rover planning. The plan specification includes activities that the operators want the rover to perform, and constraints on when they can be performed. The flight software part of the scheduler that runs on the rover, known as the onboard scheduler or onboard planner, takes this specification to generate a schedule for the rover to follow. This scheduler is able to take into account onboard knowledge, such as current temperature, data usage, and energy level. The system is also capable of flexibly executing the schedule and taking advantage of activities running longer or shorter than expected, reducing the need to build in resource margins for activities. Taken together, these capabilities will greatly enhance the productivity of the Mars 2020 rover.

2 SCHEDULER

The onboard scheduler attempts to schedule activities based on a priority determined on the ground through squeaky wheel optimization [3]. Because of limited computational resources available onboard the rover, the scheduler schedules in priority-first order, and does not move or remove activities once they are added to the schedule. Activities may have constraints such as temporal dependencies on other activities, resources to claim, state requirements and effects, and start time windows. Possible start times for the activity are determined by intersecting the valid intervals for each of the constraints together.

Within the plan specification that operators create, each activity is assigned predicted amounts of resource consumption. Resources range from data volume usage to energy usage. As the scheduler places each activity into the schedule, these resource “impacts” are tracked on timelines, which detect if any of the resource limits or constraints are violated at any time. The algorithm for computing the valid intervals of an activity scales polynomially with the number of impacts, N, as $O(N^2)$ [4]. Since the number of impacts is proportional to the number of activities, this means scheduling N activities requires running an $O(N^3)$ algorithm.

In order to achieve a reasonable runtime - on the order of a minute - for schedule generation onboard the rover, the scheduler limits itself to only considering a fixed number of activities at a time. The full plan specification may define up to 100 activities whose time windows span over multiple sols, but the scheduler will start out only scheduling the 40 highest priority ones. This is known as the “considered set”, with the rest of the activities that are skipped being in the “hopper”. As the activities in the considered set are started and marked finished during the course of execution, they are moved out of the considered set, and the next highest priority activities are pulled in from the hopper. Those activities that are finished and in the past have very little impact on the valid interval calculations for the current activities, except for inexpensive checks for dependency constraints.

The onboard scheduling problem is complicated by two factors: heating and wake/sleep/energy. We describe these challenges below.

2.1 Scheduling Rover Heating

Some activities also require parts of the rover to be heated before and during their execution (e.g. robotic arm joints, motors for driving, …). When these activities are scheduled, preheat and maintenance heating activities must be scheduled as well. Preheat activities ensure that the rover zone is heated from ambient (e.g. Mars) temperature to the temperature required for use. Maintenance activities ensure that the rover zone stays at the operable temperature as required by the activities in the schedule. When a new activity requiring a heated zone is scheduled, the scheduler can either heat the zone from ambient (preheat + maintenance) or potentially extend an existing maintenance to satisfy the heating constraints. Due to the high variability of the Mars surface temperature from day to night, the energy usage of preheat and maintenance heating activities varies significantly throughout the Martian sol. Figure 1 below shows the timing of the preheat and maintenance activities for an activity.

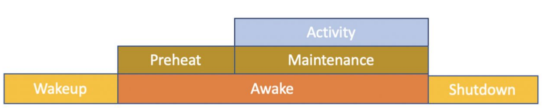

Figure 1. Activity with its required heating activities

2.2 Scheduling Rover Wake/Sleep

The scheduler must also decide when the rover should be awake and when it should be asleep. Most activities, including science activities, drives, and ground communication require the rover to be awake. When the rover is awake, the rover consumes energy faster than its power source can generate. To replenish energy then, the rover must sleep whenever it can outside of the times when it is executing these activities. This is made more difficult by constraints on the minimum duration of wake and sleep periods. Determining when the rover is to be awake and asleep thus requires a specialized algorithm [4] that carefully considers energy and activity constraints. Figure 2 shows the relationship between awake and asleep periods as well as Wakeup and Shutdown activities.

Figure 2. Activities with wakeups and shutdown scheduled for them, as well as the awake/asleep periods

At each iteration of activity placement, the scheduler considers wake/sleep constraints. It determines what times the rover must be awake due to the activities scheduled at various times and attempts to find a wake/sleep schedule that satisfies: (1) rover is awake during all activities that require the rover to be awake; (2) minimum awake and asleep minimum duration constraints; (3) wakeup and shutdown keepouts (most activities cannot occur during these periods) ; and (4) energy constraints.

2.3 Switch Groups

Additionally, the rover has limited ability to consider alternative activities depending upon resource availability using a concept called switch groups [6]. In this case the ground has specified a small number of switch groups (likely one or two) where a switch group may specify a set of activities (e.g. A1, A2, A3), with a preference to execute A3 if it fits, then A2 if A3 does not fit, or in the worst case A1. In this switch group, A3 uses more resources than A2 which uses more resources than A1. For example, A3 may be a 4x4 mosaic, A2 may be a 2x2 mosaic, and A1 may be a single image e.g. 1x1 mosaic. However, elevating a switch group (e.g.

scheduling A3) may consume resources that then prevent scheduling of other activities in the plan.

Because our scheduler is non-backtracking, the onboard scheduler cannot use search to find which level of switch group fits within available resources. Instead, the scheduler uses a Multiple Scheduler Invocation (MSI) approach, in which the scheduler is successively called with A3, then A2, then A1 until a schedule is generated with all of the requested activities (in effect a very constrained form of backtracking).

2.4 Parameter Optimization

Without backtracking, the onboard scheduler is also very sensitive to the order in which it considers activities. A change in the scheduling order can have a significant impact on whether some activities successfully get scheduled. This ordering is controlled by the "scheduling priority" parameter of the activities in the plan input. Only one static set of priorities is configured for a given plan input, and this is used to determine the schedule for potentially multiple sols.

In order to deal with this challenge, a ground system called Copilot [3] performs a Monte Carlo analysis of sol execution variability. It evaluates the performance of sets of scheduler parameters such as activity priority in order to estimate the best set of parameters for that sol/schedule. This Monte Carlo parameter optimization was shown to outperform handcrafted static strategies.

2.5 Integration with Execution

One challenge with the scheduler is that due to the limited computational resources onboard, the scheduler can only be run an expected 15-20 times per sol. Consequently, the scheduler is embedded within a flexible execution system that tolerates minor deviations from expected start times and durations, with a set of criteria that can determine when execution has varied sufficiently to warrant re-invoking the scheduler [5].

3 STATUS

The Mars 2020 onboard planner was originally scheduled to go into operations post conjunction (approximately 180 sols after landing or Fall 2021). However as this paper goes to press (September 2020), due to project schedule pressures, the Mars 2020 onboard planner is currently on hold.

4 RELATED WORK

The MAPGEN planner [7] was used in MER rover ground operations. MAPGEN was a constraint-posting planner that generated flexible time plans.

There have been a number of rover autonomy research prototypes that have demonstrated varying degrees of rover autonomy [8,9,10,11,12] of which several have demonstrated onboard (re) planning [8,11,12]. These onboard planners have all operated in far more capable onboard computing environments and therefore have not had to address the non-backtracking problem. Additionally, they typically have ignored or used much cleaner energy, wake/sleep, and preheat models.

5 CONCLUSIONS

We have described an onboard scheduler developed for the Perseverance Mars 2020 rover mission. Due to limited computation onboard the rover, the scheduler uses a non backtracking iterative approach to scheduling. The scheduler must deal with several challenges including: being embedded in execution, scheduling heating activities, scheduling wake/sleep state for the rover, and handling limited disjunction in plans with very little computation. If flown operationally, the M2020 onboard scheduler would represent a significant advance in rover mission autonomy.

Acknowledgement

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

[1] Rabideau, G. and Benowitz, E. (2017) Prototyping an Onboard Scheduler for the Mars 2020 Mission in *International Workshop on Planning and Scheduling for Space,* 142-150.

[2] Gaines, D.; Anderson, R.; Doran, G.; Huffman, W.; Justice, H.; Mackey, R.; Rabideau, G.; Vasavada, A.; Verma, V.; Estlin, T.; Fesq, L.; Ingham, M.; Maimone, M.; and Nesnas, I. (2016) Productivity Challenges for Mars Rover Operations in *International Conference on Automated*

Planning and Scheduling Workshop on Planning and Robotics (PlanRob).

[3] Chi, W.; Chien, S.; Agrawal, J.; Fosse, E.; and Guduri, U. (2019) Optimizing Parameters for Uncertain Execution and Rescheduling Robustness. In *International Conference on Automated Planning and Scheduling.*

[4] Chi, W.; S.Chien; and Agrawal, J. (2020) Scheduling with Complex Consumptive Resources for a Planetary Rover. In *International Conference on Automated Planning and Scheduling.*

[5] Chi, W.; Chien, S.; Agrawal, J.; Rabideau, G.; Benowitz, E.; Gaines, D.; Fosse, E.; Kuhn, S.; and Biehl, J. (2018) Embedding a Scheduler in Execution for a Planetary Rover. In *International Conference on Automated Planning and Scheduling*.

[6] Agrawal, J.; Chi, W.; Chien, S.; Rabideau, G.; Kuhn, S.; and Gaines, D. (2019) Enabling Limited Resource-Bounded Disjunction in Scheduling. In *International Workshop on Planning and Scheduling for Space* 2019.

[7] Bresina, J. L.; Jónsson, A. K.; Morris, P. H.; and Rajan, K. (2005) Activity Planning for the Mars Exploration Rovers. In *International Conference on Automated Planning and Scheduling* 2005 Jun 5 (pp. 40-49).

[8] Castano, R.; Estlin, T.; Anderson, R. C.; Gaines, D.; Castano, A.; Bornstein, B.; Chouinard, C.; and Judd, M. Oasis: Onboard autonomous science investigation system for opportunistic rover science. *Journal of Field Robotics.* 2007 May;24(5):379-97.

[9] Woods, M.; Shaw, A.; Barnes, D.; Price, D.; Long, D.; and Pullan, D. Autonomous science for an ExoMars Rover–like mission. *Journal of Field Robotics.* 2009 Apr;26(4):358-90.

[10] Wettergreen, D.; Foil, G.; Furlong, M.; and Thompson, D. R. Science autonomy for rover subsurface exploration of the atacama desert. *AI Magazine.* 2014 Dec 22;35(4):47-60.

[11] Gaines, D.; Doran, G.; Paton, M.; Rothrock, B.; Russino, J.; Mackey, R.; Anderson, R.; Francis, R.; Joswig, C.; Justice, H.; and Kolcio, K. Self‐reliant rovers for increased mission productivity. *Journal of Field Robotics* 2020.

[12] Ocon, J.; Colmenero, F.; Estremera, J.; Buckley, K.; Alonso, M.; Heredia, E.; Garcia, J.; Coles, A.; Coles, A.; Martınez, M.; Savas¸, E.; Pommerening, F.; Keller, T.; Karachalios, S.; Woods, M.; Dragomir, I.; Bensalem, S.; Dissaux, P.; and Schach, A. 2018. The ERGO framework and its use in planetary/orbital scenarios. *In Proceedings of the 69th International Astronautical Congress (IAC 2018).*