Abstract
The Mars 2020 Rover mission is using automated scheduling
in two capacities. As of March 2021 a ground-based sched-
uler automatically schedules rover wake/sleep and preheats to
support Mars 2020 daily operations. Included in the ground
system is an explainable scheduling tool that allows users in-
sight into the schedule developed by the automated scheduler.
An onboard scheduler that will allow the rover autonomously
better account for variations in execution is in development
for use later in the mission.

Introduction
The Mars 2020 Perseverance Rover has been operating from
Jezero Crater on the surface of Mars since February 18,
2021. This extended abstract summarizes the use of auto-
mated scheduling in the operations of the rover. A ground
scheduler, called Copilot (Yelamanchili et al. 2020), is cur-
cently used to schedule the daily operations of the rover. An
explainable scheduling tool, called Crosscheck (Agrawal,
Yelamanchili, and Chien 2020) provides explanations on
why activities failed to schedule as well as gives a visual rep-
resentation of how Copilot scheduled activities. A scheduler
onboard the rover (Rabideau et al. 2020) is under develop-
ment, with expected deployment later in the mission.

Ground Scheduling
Scheduling System
The ground scheduling system, Copilot (Yelamanchili et al.
2020), is used in the daily operations of the rover. The sched-
uler is one-shot, non-backtracking, and schedules in prior-
ity order. If a valid time for an activity is not found that
would satisfy all activity and plan-wide constraints, the ac-
tivity fails to schedule and is not included in the grounded
schedule that is returned.

Activities may require the CPU of the rover to be on
(awake). While scheduling an activity, Copilot is responsible
for generating and scheduling any necessary wakeups
and shutdowns. The rover’s power source is constantly gen-
erating energy, but the rover consumes more energy than
the power source supplies when awake. The rover’s battery
state of charge (SOC) thus only increases when the rover is
asleep. How to schedule activities and the necessary sleep
activities to ensure SOC constraints are not violated is a dif-
cult problem, discussed in (Chi, Chien, and Agrawal 2020).

Similarly to scheduling sleep activities, Copilot also gen-
erates and schedules heating activities. Activities may re-
quires areas of the rover to be sufficiently heated before they
begin, and have that heating maintained throughout the dura-
tion of the activity. When scheduling an activity, the sched-
uler also generates and schedules required preheat and main-
tenance heating activities. The duration of a preheat, and
subsequently energy use, varies with the ambient tempera-
ture, which changes throughout a sol (martian day).

Explainable Scheduling
As some activities may fail to schedule, operators would like
to understand why they failed to schedule and what changes
to the plan can be made to allow those activities to schedule.
Crosscheck, an explainable scheduling tool, provides users
with a visualization of how Copilot created a schedule and
gives information on why activities failed to schedule.

There are two main phases of scheduling an activity: the
valid intervals phase and the sleep/heat scheduling phase. In
the valid intervals phase, the valid intervals are calculated
for a subset of the activity’s constraints and are intersected
together to find times when those constraints are all satisfied.
If the intersection is empty, Crosscheck determines the min-
um subsets of constraints that did not have intersections,
and these are displayed as the reasons the activity failed to
schedule. One or more of the constraints may need to be
changed to allow the valid intervals to all intersect. If the
activity fails to schedule in the sleep/heat scheduling phase,
Crosscheck determines a more specific reason. Failure rea-
sions during this phase include there not being enough en-
ergy available at the time the activity was attempted to be
scheduled at, or that a necessary heater is not allowed to be

Copyright © 2021, California Institute of Technology. U.S. Gov-
ernment sponsorship acknowledged.
Figure 1: Crosscheck allows users to view the schedule and specific information on why an activity failed to schedule as well as which constraints may need to be adjusted.

Figure 1 shows an example of a plan displayed in Crosscheck. The highlighted activity failed to schedule because there were no times when both the activity’s dependency constraint and the execution constraint were satisfied. For further details on how Crosscheck calculates why activities failed to schedule and all the information it displays, see (Agrawal, Yelamanchili, and Chien 2020).

**Onboard Scheduling**

A scheduler onboard the rover is in continued development (Rabideau and Benowitz 2017; Rabideau et al. 2020). The use of a scheduler onboard the rover seeks to improve the productivity of the rover, by taking advantage of available resources if activities finish early or use less resources than expected. Rather than uplinking a set of sequences with fixed times the rover, a plan file will be sent and the onboard planner will schedule and execute activities. The challenges with embedding a scheduler in execution, such as when to reschedule and how to handle events that occur during scheduling are discussed in (Chi et al. 2018).

Copilot contains the same core scheduling algorithm as the onboard scheduler, and also contains an execution simulation capability. This allows Copilot to predict how the onboard scheduler will behave on a given plan, and optimize parameters for the onboard scheduler to use (Chi et al. 2019).

**Acknowledgments**

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

**References**


