Pathogen: Using Campaign Intent To Guide Onboard Planning For A Self-Reliant Rover

Joseph A. Russino and **Daniel Gaines** and **Steve Schaffer** and **Vincent Wong**
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109
joseph.a.russino@jpl.nasa.gov

Abstract

Previous case studies of the MSL Mars rover mission identified onboard activity planning as a key capability to improve productivity of planetary surface exploration missions. NASA's transition to non-sun-synchronous relay communications at Mars further magnifies the need for reduced reliance on ground-in-the-loop interaction with future rovers. A new prototype onboard planner called Pathogen was implemented to explore the challenges of effectively conveying operator intent, efficiently producing high-quality plans, and responsively adapting to both challenges and opportunities encountered during execution. In particular, the onboard planner can re-route the mission path as terrain data accumulates as well as incorporate updated objectives provided by in-situ autonomous science analyses. Preliminary field evaluation of the Self-Reliant Rover system by planetary scientists using an MSL-class research rover on Earth to conduct a walkabout science camapign indicated significant increases in mission productivity: 80% decrease in required mission duration and 267% increase in total locations surveyed.

Introduction

Achieving consistently high levels of productivity has been a challenge for Mars surface missions. While the rovers have made major discoveries and dramatically increased our understanding of Mars, they require a great deal of interaction from the operations teams, and achieving mission objectives can take longer than anticipated when productivity is paced by the ground teams' ability to react. We have conducted a project to explore technologies and techniques for creating Self-Reliant Rovers: rovers that are able to maintain high levels of productivity with reduced reliance on ground interactions.

A full overview of the Self-Reliant Rovers (SRR) design has been supplied in (Gaines et al. 2018). In this paper, we provide a more complete and detailed discussion of the onboard planning problem, and present the results of a series of component-level and system-level evaluations that we performed. For context, we begin here by first reviewing two elements of that previous discussion: the specific challenges gleaned from the planetary rover case study that motivates this effort, and our formulation of *campaign intent*, which forms the basis on which we define our planning problem.

We conducted an extensive case study of Mars Science Laboratory (MSL) operations in order to identify significant productivity challenges (Gaines et al. 2016). The case study included an analysis of the challenges the team faced in making more effective use of time and vehicle resources. Two of the main challenges, which we address in part by the inclusion of an onboard planner, are predicting vehicle resource usage and strict reliance on ground-in-the-loop for target selection and drive planning.

It is difficult to predict how long activities will take to complete. Operators tend to overestimate duration to avoid activities being cut off. As a result, activities typically end earlier than expected which contributes to rover idle time. In addition, the rover relies on ground operators to pick out specific science targets and to identify paths around slip hazards, such as patches of loose sand. This results in a significant drop in productivity on sols (Mars solar days) that follow drives during constrained periods of the mission. Even during non-constrained sols, it constrains the timing of activity that can change the state of the vehicle and activity that acquires decisional data to occur prior to the decisional communication pass.

In this paper, we highlight the design considerations and a prototype implementation of an onboard *goal planner* that we developed to address these challenges. In particular, we present the Pathogen planning algorithm, which we developed to provide high-quality plans that meet the needs of this particular effort. Via the inclusion of an onboard planner, our proposed system is able to incorporate up-to-date knowledge of onboard resource levels and vehicle state to generate sequences of activities that fulfill high-level mission objectives. This allows the team to use less conservative modeling of activity resource use and duration, which in turn contributes to less idle time due to unused margin. The planner also enables the system to respond to new objectives identified by an onboard autonomous science subsystem (described in detail in (Gaines et al. 2018)).

Under our proposed approach, ground-based planning teams retain the ability to command specific actions, but the primary means of guiding rover operations becomes the crafting of these high-level goals. In this way, the onboard goal planner supplements and enhances, rather than

©2019. California Institute of Technology. Government sponsorship acknowledged.

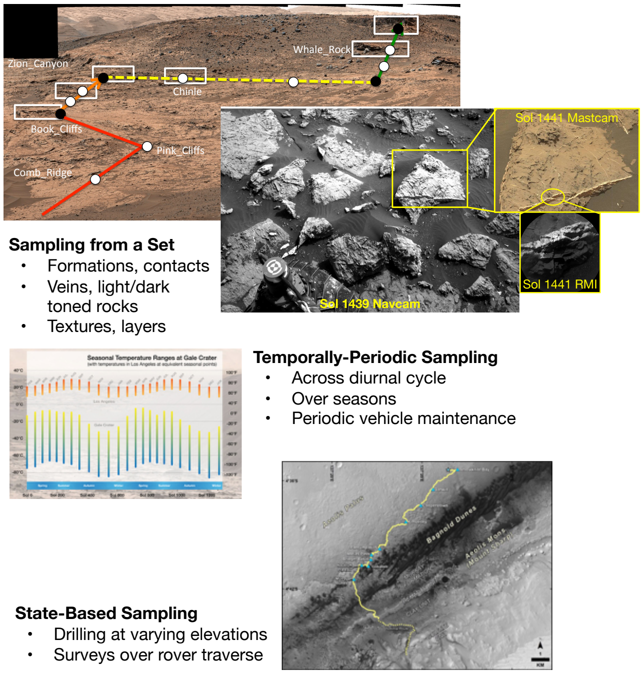


Figure 1: Summary of Campaign types.

replaces, the traditional tactical planning process.

Campaign Intent

A significant challenge to maintaining high rover productivity under reduced operator interaction is conveying operator guidance and objectives without requiring operators have up-to-date knowledge of the rover and its environment. In traditional operations, each planning cycle begins with a review of the current long term objectives of the mission presented in the context of the latest available rover state data (Chattopadhyay et al. 2014). The operators assimilate all the various objectives, state data, and mission knowledge in order to synthesize a high quality plan that makes progress toward the goals while respecting limited rover resources such as time, energy, and data volume.

The team will typically have several high-level objectives to pursue, as well as a variety of supplementary objectives. A wide range of recurring engineering activities also need to be accommodated: instrument calibrations, telemetry collection, system configuration management, etc.

Importantly, the quality of the plan is not just a function of what activities are scheduled; it depends on how well they relate to the current objectives and to each other.

We developed the concept of *campaign intent* to convey such information to the rover so that it may generate its own prudent in-situ plans when human guidance is prohibitively delayed. Campaign intent specifies a set of *Goals* for the rover and *Campaigns* which define the relationships among those Goals. We gleaned three initial types of Campaigns from MSL scenarios, as summarized in Figure 1:

Class sampling: Choose observation targets that best exemplify a particular feature (e.g. geological layering). Once identified, the targets form a Goal Set. The plan typically accumulates utility as additional targets are included, but eventually reaches a point of diminishing returns.

Temporally-Periodic sampling: Schedule Goals to match a repeating temporal pattern (e.g. hourly). The preferred Goal cadence typically allows at least some timing flexibility.

State-based sampling: Trigger Goals based on the evolution of the rover/terrain state (e.g. at every 50m traveled). The state criteria is typically expressed as a preferred cadence with some flexibility.

For prototyping purposes, we identified and implemented a few specific Goal types that were chosen to represent common kinds of operational objectives seen on rover missions, while also being amenable to entirely autonomous execution by the available research rover hardware. Goals describe how they award utility to a plan, but might also be able to generate concrete options for extending the plan.

Target Goal: Rewards proximity to a target location and performance of science observations there. Suggests drive actions toward the target (possibly broken up with stops for interrupt events) and then execution of the science task on arrival.

Cyclic Goal: Rewards scheduling some action with a desired cadence. Suggests executing the action when next cadence timepoint arrives (possibly as an interrupt event).

Limit Goal: Rewards rover state that remains within the resource limits. Suggests actions that help move away from the limits (possibly as an interrupt event, e.g. sleep periods to allow battery regeneration).

Dimension Goal: Rewards increasing certain resource values (e.g. distance traveled).

Conflict Goal: Penalizes incompatible states (e.g. overlapping sleep and drive actions).

Problem Definition

Inputs

Operating with this notion of campaign intent, we frame the onboard planning problem as follows:

The rover is provided with a set of Goals. Each Goal has an associated *Utility* value and strict *Priority* rank, with higher Priority always overriding any combination of lower Priority, but equal Priorities competing based on their Utility. In addition, operators may define Campaigns that award additional Utility to a plan if specific combinations of Goals are achieved. The two main Campaign types are *Periodic Campaigns* and *Goal Set Campaigns*. Periodic Campaigns award Utility when two or more activities of a given type are scheduled at a desired sampling cadence, whether temporal (e.g. once per hour) or state-based (e.g. once every N meters of travel). Goal Set Campaigns award Utility when the plan contains a number of activities of a particular type within some specified bounds, useful for class sampling scenarios. For example, a Goal Set Campaign might award 10.0

Utility after the rover takes 3 images of geological layering, plus 2.0 more Utility for each additional such image up to a maximum of 5.

While attempting to satisfy the provided Goals and Campaigns, the planner must obey a set of provided engineering constraints, including energy limits, instrument heating requirements, and required battery state of charge at a designated handover time. It must also accommodate exogenous events such as fixed communication windows and the day/night cycle. Constraint violations result in a large negative Utility penalty.

The planner is also provided the starting state of the system and a model of how each parameterized activity instance impacts the evolution of those states. The modeled states include: location of the rover, battery state of charge, instrument heating states, and the lighting state. In some cases, resource and state modeling may be provided by other rover subsystems. For example, in our prototype implementation the expected duration and energy expenditure for a planned drive activity is provided by an onboard navigation path planner, using up-to-date local terrain maps.

Outputs

Based on the above inputs, the planner searches for an optimal plan - i.e., a sequence of activities which achieves as much Utility as possible without violating any of the engineering Constraints. The resulting plan must include not only activities which directly contribute Utility by satisfying Goals and/or Campaigns, but also those activities which are necessary to maintain compliance with the engineering constraints, such as pre-heating of instruments, sleeping to allow state of charge to regenerate, and pre-planned communication activities.

The planner must execute in real-time on rover hardware and provide the best plan available whenever its time-slice expires. In addition, it must be able to re-plan at any time. Re-planning may be triggered by the arrival of new Goals (whether from human operators or from other rover subsystems such as automatic science data analysis) or by the significant divergence of some rover state from the predicted/expected state of the system (e.g. a drive that takes too long).

Pathogen: Using Campaign Intent to Guide Planning

Overview

Each Goal in our system implements a *successor generator* which takes in an existing plan and moves the state forward, typically by adding activities to the plan which achieve or make progress toward the Goal. Goals that specify activities which must be performed at specific times may also implement an *interrupt requester*, which marks points in the proposed plan where the Goal would like a chance to add a successor.

Our approach to plan generation is based on branch-and-bound search, is outlined in Algorithm 1. Nodes in the search graph represent partial action plans and a corresponding forward-sweeping "current state". A node's score is the sum of its past Utility (i.e. the Utility of all Goals which are satisfied by the scheduled activities) and a heuristic "future" Utility. Disincentivization of state conflicts is accomplished by attributing large negative Utility to such conflicts when calculating a node's score. Utility awarded for satisfying the criteria of Campaigns, as described in the previous section, is also taken into account when calculating a node's score.

Result: Valid plan
Generate seed nodes
while *! Done* **do**
    Pick best pending node
    Expand that node
    Update pending nodes
end
Pick best pending node

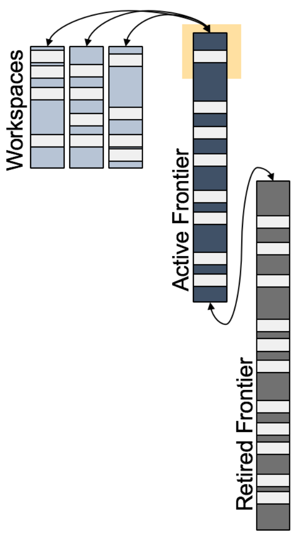
Algorithm 1: Pathogen Search

Starting from the empty plan, each iteration of search expands a chosen partial plan into many possible successor plans (the branches). Each potential successor is scored and must exceed a running threshold of plan quality (the bound) in order to be retained for future expansion; otherwise it is pruned (along with all its descendants). Specifically, the optimistic maximum quality of any plan based on the candidate partial plan must exceed the pessimistic minimum quality prediction of all other candidates already considered. The frontier of un-expanded partial plans is periodically sorted by estimated final plan quality, yielding a hybrid of depth-first and best-first expansion order.

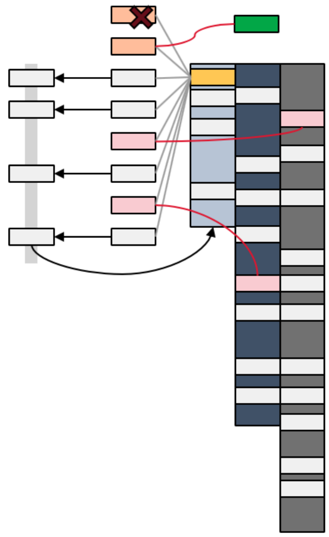
Node Expansion

As illustrated in Figure 2a, We take a multi-threaded approach to expanding nodes. Pathogen maintains a (per-thread) *workspace frontier*, which grows in depth-first bursts and is periodically merged back into the main frontier. An *active frontier* repository is sorted by heuristic value. Threads check out nodes from the active frontier, and then merge the expanded nodes that they've generated back into it. For efficiency, we limit the size of the frontier by collecting the least promising nodes into a separate *retired frontier*. These nodes may be called back into the active frontier as needed (on underflow, or languishing active).

When a node is to be expanded, we first check its score to determine if it is prunable. If it could not beat the current best node even if it achieved its full maximum possible future Utility, we drop it. Given that we allocate large negative Utility for nodes containing conflicts (with exogenous events, or with resource constraints), nodes with such conflicts are always removed in this step. For nodes which are not pruned, we generate successor nodes by appending new activities to the plan. These successor nodes are scored, and any poor-quality children are immediately pruned, as are any which duplicate previously-examined nodes. The latter is achieved using a hash function on the plan contained within the node. This leaves only those child nodes which are conflict-free and which have potential to beat the best plan discovered so far. Those resulting successor nodes are used to update the



(a) Nodes are expanded in one or more (per-thread) workspaces. Low-scoring nodes are retired from the active frontier, which is maintained in a sorted state.



(b) Low-scoring and duplicate successor nodes are pruned. Remaining successors are added back into the workspace queue, or directly into the active frontier if they can be expanded no further.

Figure 2: Node Expansion and Successor Generation in Pathogen

best node metrics and are pushed either into the workspace (for further expansion) or directly into the active frontier (if they cannot be expanded further).

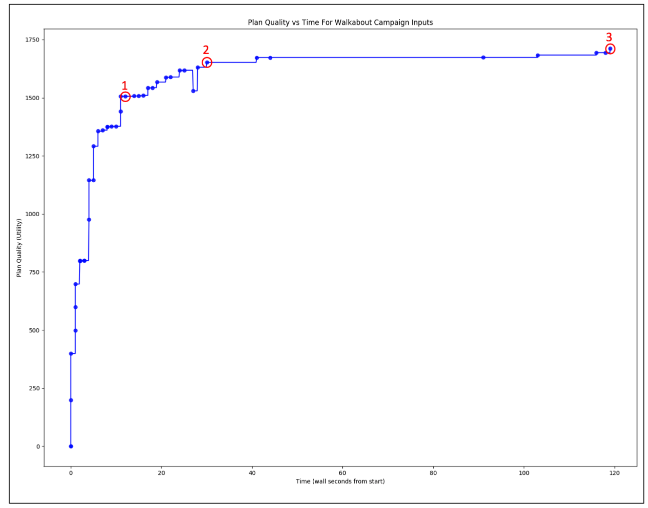
Successor nodes are generated from an existing node by looping over all remaining Goals and generating activities that make progress toward satisfying these Goals. Partial plans are always expanded forward in time by appending one of the possible subsequent activities to the growing plan. Long activities may be broken up into segments, if possible, to allow for interrupts. We also check for opportunities to skip ahead in time, in particular to the ends of upcoming exogenous activities and to future times that correspond to goals specified by time cadence. Finally, predetermined exogenous events are also inserted at the appropriate times during the successor generation step. This process is illustrated in detail in Figure 2b.

Performance And Trade Offs

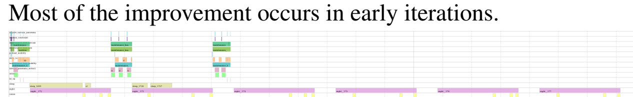
The complete search can be very time intensive, but is guaranteed to return an optimal plan according to the expressed campaign preferences. Pathogen is an anytime algorithm (Boddy 1991); even without running to completion, the search can return the best plan encountered so far. This feature allows the rover to limit its planning time and proceed to be productive with a reasonable (but not provably optimal) plan. Minor plan perturbations during execution are accommodated by time-efficient repair strategies (for example, to shift actions forward after a small driving delay), while major disruptions (such as an insurmountable obstacle in a drive, or the injection of an entirely new goal) invoke a full replanning cycle so that all goals are reconsidered.

Evaluation

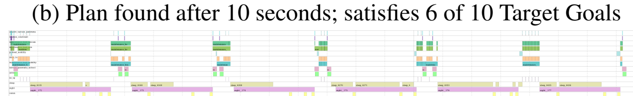
Onboard Planning Performance Evaluation



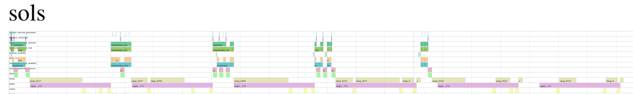
(a) Goal Planner finds higher-quality plans the longer that it runs. Most of the improvement occurs in early iterations.



(b) Plan found after 10 seconds; satisfies 6 of 10 Target Goals



(c) Plan found after 30 seconds; satisfies all 10 Target Goals in 7 sols

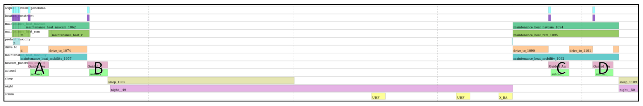


(d) Final plan found after 2 minutes of search; satisfies all 10 Target Goals in only 5 sols

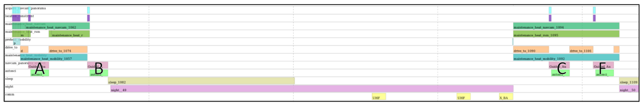
Figure 3: Plan Quality vs Time and Sample Intermediate Plans

Figure 3 demonstrates plan quality improving over time during the search process, highlighting the anytime nature of Pathogen. In this particular example, the goal planner has been provided with the goals of performing Autonomous Science activities at 10 different target locations and a planning horizon long enough to allow for accomplishing all of them. Early iterations result in plans that accomplish only a subset of the goals (Figure 3b). Within 30 seconds it has found a plan (Figure 3c) that accomplishes all 10 goals within the 7-sol horizon. By the end of the allocated runtime of 2 minutes, the planner has found a more efficient plan that accomplishes all 10 objectives over the course of only 5 sols (Figure 3d).

In many cases we will want to ask our system to generate high-quality plans while over-constrained. When the



(a) 2 sol planning horizon, 5 goals (A-E)



(b) 2 sol planning horizon, 6 goals (A-F). The planner achieves a higher total score by replacing the activity at target D with a higher-Utility activity at target F

Figure 4: Over-constrained 2-sol plans. Targets A, C, D, and E were assigned Utility values of 100, while targets B and F were assigned Utility values of 200

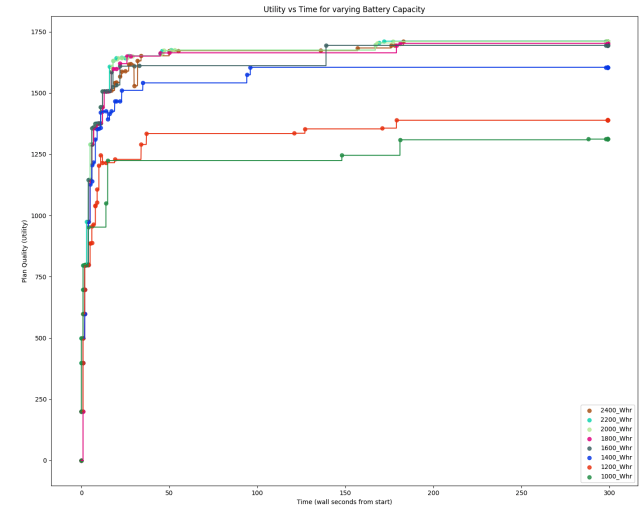
amount of time or resources available makes it impossible to accomplish all of the provided goals, our system needs to intelligently choose which to perform and in what order.

Utility values for each individual goal can be used to express their relative importance to the planner. In the set of tests illustrated by Figure 4, we restricted the planning horizon to 2 sols and provided more goals than could be accomplished within that time. In Figure 4a, the input goals correspond to targets labeled A-E.. All of these goals have a Utility value of 100, except for the goal at location B, which has a Utility value of 200. In the resulting plan, the rover drives first to target A, then B, C, and finally D. Target E is too far to reach within the remaining amount of time, but the rover is scheduled to start driving toward it at the end of the second sol. In Figure 4b, we add a goal at target F that has a Utility value of 200. In the resulting plan, the rover will drive from target C to F in the second sol, instead of finishing at target D as it had in Figure 4a.

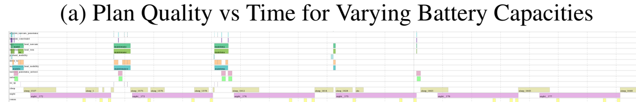
Similarly, the goal planner is capable of producing high-quality plans while honoring constraints on resource usage. In Figure 5, we show plan quality and sample plans for varying rover battery capacity values. With a lower-capacity battery, the rover needs to sleep (to recharge) more frequently and can therefore satisfy fewer goals within any given amount of time. In this way, lower energy resources typically result in fewer goals achieved and thus lower-Utility plans. For example, the rover resource model used for Figure 3 included a 2400 Whr battery; in that case, the planner was able to schedule activities to satisfy all 10 goals with additional sols to spare. In contrast, the plans presented in Figures 5b and 5c show the rover accomplishing only 5 and 7 of its goals, respectively, and spending a larger proportion of the schedule sleeping. This illustrates the flexibility of our approach to handling a wide variety of resource constraints.

System Evaluation: Mars Yard Walkabout Campaign

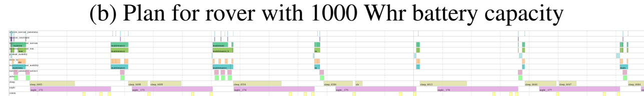
We have developed a prototype implementation of the Self-Reliant Rover approach using the Athena research rover. In order to evaluate the ability of the Self-Reliant Rover approach to increase productivity we conducted a simulated *walkabout* campaign in which actual planetary scientists used our system to explore a geographical region. A walkabout is a reconnaissance campaign in which operators command the rover to make an initial pass over a region of interest performing remote sensing observations The data collected during a walkabout is then used to decide which locations to revisit for more in-depth study. We selected a walkabout campaign for the evaluations because it has been found to be an effective means of exploring a region of interest (Yingst et al. 2017) and is anticipated to be used in the Mars 2020 mission to help identify sampling locations.



(a) Plan Quality vs Time for Varying Battery Capacities



(b) Plan for rover with 1000 Whr battery capacity



(c) Plan for rover with 1400 Whr battery capacity

Figure 5: Plan Quality vs Time and Sample Plans for Varying Battery Capacity

Our primary objective in conducting the simulated walkabout campaign was to evaluate the ability of the SRR approach to enable productivity for rover operations with reduced ground-in-the-loop interactions. The productivity metrics we used are based on the MSL case study described in (Gaines et al. 2016). These metrics relate to how long it takes to accomplish campaign objectives:

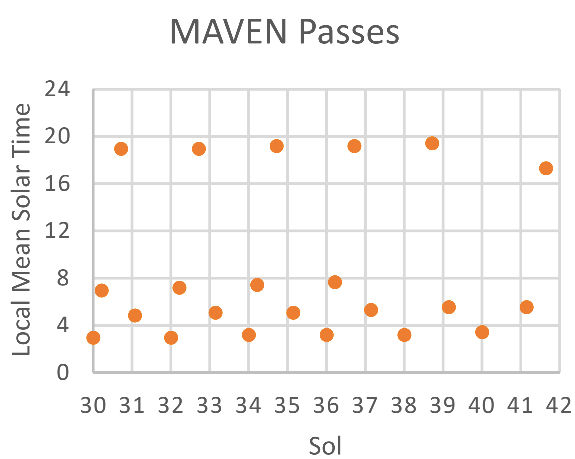
- Percentage of sols making significant contributions toward campaign
- Number of sols to complete objectives
- Number of locations surveyed during campaign

For our region of interest, we constructed geological scenes in the JPL Mars Yard. The area that we created can be seen in Figure 7. We simulated a larger area by applying an

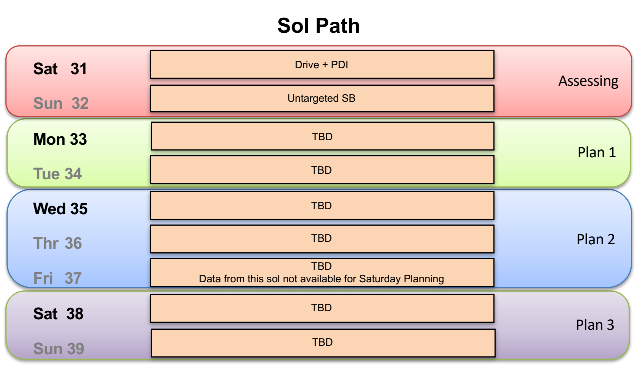
8x scaling factor to the actual Mars Yard dimensions. This allowed us to simulate a longer-duration mission than would otherwise be feasible in the Mars Yard. We similarly scaled time between the simulated mission and actual rover activity to match realistic activity durations from MSL operations.

Three planetary scientists from the MSL mission participated in the evaluations. We prepared strategic guidance for the scientists similar to the guidance they would be provided for an actual campaign on Mars. This included the labeled imagery in Figure 7 showing units and features identified from "orbital" data. The team was also provided with contextual information and strategic guidance for the campaign.

Because we are interested in the impact of non-sun-synchronous orbiters, we used a projected overflight pattern based on the MAVEN (Mars Atmosphere and Volatile EvolutioN) orbiter. Figure 6a shows the timing of projected orbiter overflights that were used for the simulated campaign.



(a) Projected relay orbiter overflight pattern.



(b) High-level sol path.

Figure 6: A realistic sol path was generated for this campaign using projected overflight pattern from the non-sun-synchronous MAVEN orbiter.

We used the overflight information from Figure 6a to define the beginnings of a "sol path" for the campaign. A sol path is a high level summary of the team's near-term plans. It groups the upcoming sols into groups based on which sols will be planned together in a multi-sol plans. A complete sol path would also state the high level activities anticipated for each sol. We left that description out, leaving it up to the scientists to decide how to spend each sol. Figure 6b shows the sol path we provided for the Mars Yard Walkabout Campaign.

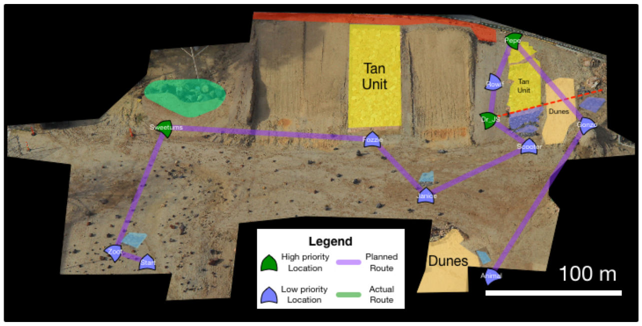
We met with the scientists on three separate occasions in the course of conducting the simulated mission. At our first meeting, we provided them with high-level guidance as previously described and then collected their objectives for the first phase of the walkabout campaign (sols 33 and 34), encoded formally as Goals and Campaigns. In the second and third sessions, we presented the results of the previous plan's execution, asked for new inputs, and solicited feedback.

Between each of these meetings, we uploaded the scientists' inputs to the research rover and allowed it to plan and execute accordingly. It should be noted that our system is still in a prototype stage in which not all bugs have been worked out. However, we wanted to conduct a fairly ambitious evaluation of the system in this relatively early stage to help guide future development. To facilitate this evaluation, we developed a "checkpointing" capability that allowed us to save a snapshot of the state of execution and resume execution from a previous checkpoint. This enabled us to restart execution from saved checkpoint if a problem was encountered without having to restart from the very beginning. As such, the results that follow represent a composite of executions.

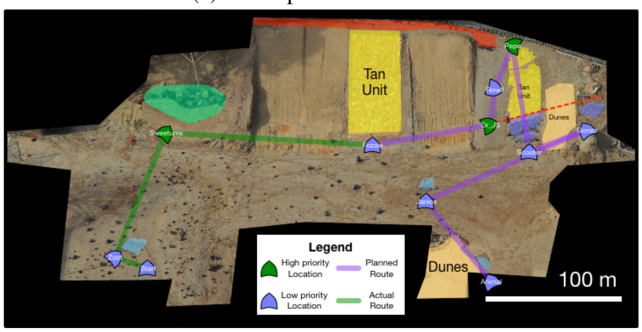
Figure 7 shows the initial planned route and final as-executed route for each of the three execution sessions. While the rover was able to explore all 10 locations of interest chosen by the scientists, the order in which it eventually visited these locations was different from the order in which it initially planned to visit them at the start of the mission. The system re-planned a number of times over the course of the walkabout, as a result of new inputs from the ground team, new follow-up observations suggested by the onboard autonomous science subsystem, and unanticipated state changes resulting from unpredictable execution. In the latter case, we found that our system needed to frequently replan during or following long drives, as the onboard navigation system refined its estimate of how long it would take to drive between two target points and potentially routed around previously-unseen hazardous terrain.

We met with the scientists for one final session to evaluate the results of the walkabout campaign. In order to determine how well the rover performed in the walkabout we asked the scientists to review the results of each visited location and assess if the location had been sufficiently surveyed to meet the campaign objectives. While there were cases in which the scientists would have selected different observations from those selected by the rover, they concluded that each location had been sufficiently surveyed. Further, they concluded, given the locations that were visited, that the walkabout had successfully achieved the strategic objective of surveying the Mars Yard region.

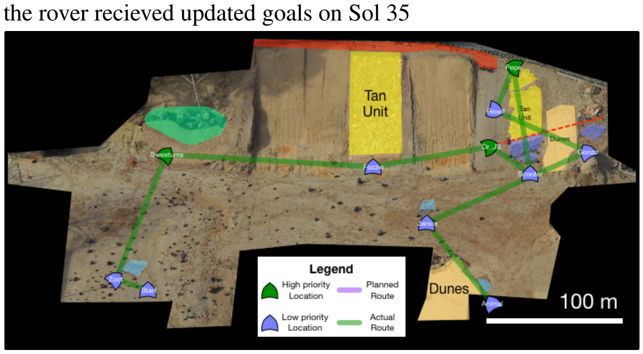
As a baseline for performing a productivity comparison,



(a) Initial planned traversal



(b) The traversal order was updated mutiple times over the course of the campaign. Here we see an intermediate plan produced after the rover recieved updated goals on Sol 35

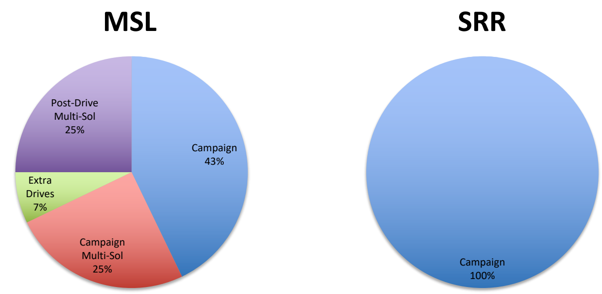


(c) Final path traversed

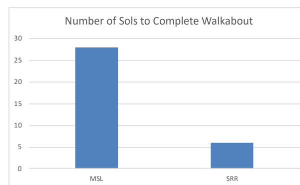
Figure 7: Planned traversal order evolved based on new goals and new information acquired over the course of the 7-sol walkabout

we estimated how the campaign would have been conducted using an MSL-style rover. For each ground-in-the-loop cycle, an MSL-style rover would be limited to surveying at most one location, due the reliance on operators selecting observations, and driving to the next location. For drive segments that required the rover to avoid sand, an MSL-style rover would require additional ground-in-the-loop cycles as it would rely on human operators to identify the sand hazards and plan paths around them.

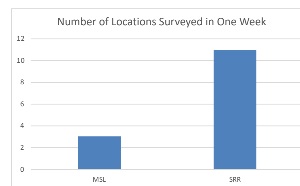
Figure 8 provides quantitative measurements of the productivity improvements achieved by the Self-Reliant Rover approach. In Figure 8a we provide a breakdown of sol productivity similar to the ones we performed for the MSL case study in (Gaines et al. 2016). The MSL approach was projected to have 32% low productivity sols due to the need to wait for ground-in-the-loop for performing location surveys and to guide the rover around areas with sand hazards. In contrast, the SRR approach was able to make significant progress toward campaign objectives on each sol. This results in SRR achieving a 47% increase in productive sols.



(a) Comparison of sol productivity: SRR showed a 32% increase in sol productivity.



(b) SRR achieved 80% reduction in sols to complete campaign.



(c) SRR achieved 267% increase in locations surveyed in one week.

Figure 8: Quantitative evaluation of Self-Reliant Rover productivity.

Figure 8b compares the number of sols required to survey all of the locations selected by the scientists. An MSL-style rover would require 28 sols to perform the survey while the Self-Reliant Rover completed the campaign in 6 sols. This represents an 80% reduction in sols to complete the campaign.

Finally, in Figure 8c we compare the number of locations the rovers would be able to survey if we were to restrict the walkabout to a single week. An MSL-style rover would survey only 3 locations while the Self-Reliant Rover surveyed 11 locations. This is a 267% increase in number of locations surveyed.

Over all, the walkabout campaign demonstrated that the Self-Reliant Rover approach is able to maintain high levels of productivity with limited ground-in-the-loop cycles. The approach provided mechanisms that allow the scientists to effectively guide the rover's behavior despite the limited communication opportunities.

Discussion and Future Work

Planner Expressivity

While the notion of Campaign intent and the underlying search-based planner's notions of Utility and Priority provided a powerful interface for expressing intent to the planner, we did identify a number of challenges related to effectively guiding the planning process.

Oversubscribing, or providing more Goals than the rover can perform given its time and resource constraints, tends to result in shedding lower Priority objectives; however, this isn't always desirable if the rover will have the opportunity to accomplish those Goals in the next planning cycle. One potential approach, which was employed during our mock walkabout execution, is to extend the planning horizon beyond the next uplink, perhaps to the limit you will allow rover to accomplish a given set of objectives. The trade-off inherent in this approach is that the longer planning horizon may result in longer planner running times.

The scientists participating in the evaluations suggested additional options for how to specify Goals to the rover. Our interface required users to specify the location the rover should drive to and heading it should point in order to characterize a location. Instead, the scientists wanted to specify what area to characterize and have the rover determine the appropriate location and heading.

The scientists also would have preferred the ability to specify locations the rover should be at certain times. For example, if the team knows there will be a good ground-in-the-loop opportunity, they may want the rover to be at a particular location so that they can better direct the characterization of that area.

Related to priorities, the scientists suggested that the planner might take into account objective Priority when deciding on the *order* in which locations are visited. If it does not result in significant increase to drive duration, they would prefer higher Priority locations be visited earlier. This would allow the team to receive the data earlier, giving them more time to analyze the data. More generaly, there may be value in taking into account more than just the Goals that are accomplished and their relative importance when scoring individual plans. One might also want to penalize, for example, excess drive distance, mast slew, or idle time.

Planner Design & Implementation

We found that the depth-first approach allows Pathogen to very quickly reach complete plans that use up the whole horizon. Choosing the best of the successors to expand next, in a limited best-first fashion, results in generally busy plans that achieve many objectives. And allowing the search to occasionally jump to the globally most promising node helps brings it out of a local maximum.

However, there are still issues that remain with the over all search algorithm. One of those issues is the heuristic functions Pathogen uses for node evaluation. The heuristic defined for each Goal and Campaign tended to vastly overestimate the Utility that can be achieved, due to their ignorance of other input objectives:

Given a set of Target Goals, the heuristic function for each one only evaluates whether the rover has the time left to travel to and achieve itself. This is admissible, but results in the rover planning suboptimal paths due to an expectation that it can achieve all of the Goals, when the more accurate expectation is simply that it can achieve at least one of any of them.

Cyclic Goals direct the rover to stop every X meters during a drive and perform imaging. Each imaging instance doles out a Utility value. The heuristic function for this Campaign cannot predict how much the rover will actually drive in the future, so it makes the only admissible assumption that the rover will always be driving. This ends up greatly overestimating the achievable Utility.

In general, the independent reasonable heuristic and successor behavior of each Goal may result in potentially undesirable global search behavior. Much of our initial design was in figuring out what and how to express science objectives and operations in terms of Goals and Campaigns. How to balance their interactions when translating to node evaluation and successor generation turned out to be the most difficult part of the implementation. To mitigate the Target Goal issue, for instance, we might have instead considered the targets together as a set that internally chooses a traversal order using a simpler path planning algorithm. Alternatively, we might have introduced another primitive Goal type that rewards plans for choosing shorter paths.

We also learned that using the Priority & Utility as the sole metric for evaluating a node is insufficient to express most constraints. Many factors can result in invalid plans; for instance, a plan that skips mandatory, user-specified activities like Comm windows, or a plan where the state of charge goes below the allowed minimum, should not be allowed. Such constraints cannot be properly expressed with Utility, and we relied on encoding external knowledge that these nodes, failing these conditions, should be pruned. Because these constraints are evaluated as binary conditions, the search cannot be guided away from violating them by heuristics, only backtracking once it realizes the constraint has been violated.

Reserving Resources for Future Activity

When the planner generates plans that include autonomous science activities, it does not know ahead of time how many follow-up Goals will be proposed by the autonomous science algorithms. This makes it difficult to appropriately allocate resources in the plan. The planner needs to reserve resources for these future autonomous activities to enable follow-up observations to be performed. However, if the planner reserves too much, then this may unnecessarily limit the amount of activity that can be performed earlier in the plan.

The approach we took with the current implementation was to have the autonomous science activity that represents running the detectors include additional resource reservation to account for a set amount of follow-up observations. This forces the planner to set aside resources for follow-up observations whenever it plans an autonomous science activity. Because the activity does not actually consume the reserved resources, there will be available resources for the planner to make use of when re-planning to accommodate the newly proposed follow-up observations.

Planning and Execution

It is often the case that the execution of an activity depends on the successful completion of the activity that precedes it.

We found that it was crucial that the state of the system and the progress and ultimate success/failure of activities were monitored and communicated back to the planner in real-time. This allowed us to update the planner's internal model of system resources so that future invocations would have up-to-date information, but more importantly it also allowed us to dynamically trigger re-planning if the state of the system deviated too far from what the planner had anticipated when the plan was generated. Future work should consider this re-planning based strategy alongside other methods of handling variability in execution, such as dynamically shifting and extending/contracting activity timing.

Some activities, particularly long-running activities such as drives, may need to be broken up into smaller chunks in order to accommodate the timing of other activities - comm windows, periodic observations, sleep, etc.. In these cases, our system generates the breaks in the long-running activity automatically, and therefore needs to fill in the success criteria automatically based on the requirements of the immediate successor activity.

One particular execution-related issue that we have not addressed in this work is how to deal with activities that continually fail. For example, a particular Goal location may not be reachable due to traversability issues (e.g. the target is the middle of a sand bowl or blocked by rock hazards). The strategy of re-planning and re-scheduling may not help in this case. There is room for future work in determining when a failed activity indicates that a particular Goal cannot be accomplished and should be removed from consideration in the planner.

Related Work

Shalin, Wales, & Bass conducted a study of Mars Exploration Rovers operations to design a framework for expressing the intent for observations requested by the science teams (Shalin, Wales, and Bass 2005). Their focus was the use of intent to coordinate planning among human operators and the resulting intent was not captured in a manner that would be conducive for machine interpretation. Our approach codifies some of the fields in their framework in a way suitable for the rover. In particular, the authors defined a "Related Observations" field as a way for scientists to identify relationships among different observations, which need not be in the same plan. Our work on campaign intent can be seen as a way of defining a specific semantics to these types of relationships to facilitate reasoning about these relationships by the rover.

Their framework also includes information that we agree is essential for effective communication among operators but that we do not currently express to the rover. For example, the "Scientific Hypotheses" field is used to indicate what high-level campaign objective is being accomplished by the requested observation. We are not yet providing these higher-level campaign objectives to the rover, though it is an interesting area of future research.

Mali views intent as a means for a user to place constraints on the types of plans a planner is allowed to produce such as only generating plans that have at most one instance of a class of actions or that plans must limit the use of a particular action (Mali 2016). The primary role of our use of intent is to allow the planner to assess the value of achieving a given set of goals. However, some of our campaign intent does imply constraints and preferences on how, or more specifically, when goals are accomplished. For example, the periodic campaign intent specifies a timing relationship among goals and a preference on how close to comply with the desired timing.

There are some similarities between our campaign definitions and those used for Rosetta science planning (Chien et al. 2015). Both use campaigns to express requests for variable-sized groups of observations with relationships and priorities. Rosetta plans covered much longer time periods (e.g. weeks) and required more complex temporal patterns, such as repeating groups of observations. But observation patterns were primarily driven by the predictable trajectory of the spacecraft, allowing relationships to be expressed as temporal constraints. This is not sufficient for rovers, where many observations are dictated by the rover location and surrounding terrain, and the duration of many activities cannot be accurately predicted. State-based and goal set relationships more accurately represent some of the science intent found on surface missions.

There have been several integrated rover systems with similar objectives to our work including PRoViScout (Paar et al. 2012), Zoe (Wettergreen et al. 2014), and OASIS (Castano et al. 2007). These systems include autonomous science capabilities to enable onboard identification of science targets. Similar to our approach, they select follow-up observations for identified targets and submit these requests to an onboard planner to determine if there are sufficient resources to accomplish these new objectives. OASIS uses the CASPER (Continuous Activity Scheduling, Planning, Execution, and Re-planning) continuous planner (Chien et al. 2000), like SRR, but relies on a simpler iterative repair algorithm where we use Pathogen for plan generation. The campaign intent concepts we have developed would also be applicable to PRoViScout as a way to increase the expressivity for providing scientist intent to the rover.

There have also been a variety of autonomous science systems deployed or proposed for rovers including the AEGIS system running on the Opportunity and Curiosity rovers (Francis et al. 2017), and the SARA component proposed for an ExoMars rover (Woods et al. 2009). These systems allow the rover to identify targets in its surroundings that match scientist-provided criteria. The introduction of campaign relationships broadens the scope of the type of guidance that scientists can provide these systems, allowing scientists to express the amount of observations they would like for their different objectives along with the relative priorities of the high-level objectives.

Automatic goal generation has precendent in systems such as ARTUE (Klenk, Molineaux, and Aha 2013). In thier Goal-Driven Autonomy framework, new goals may be automatically generated in response to state discrepancies detected during execution. These generated goals are typically intermediate objectives in a Heirarchical Task Network, whereas in our system we allow for entirely new top-level objectives to be submitted to the planner by other autonomy subcomponents.

The Mars 2020 mission is planning to incorporate onboard scheduling to improve resource utilization of the rover (Rabideau and Benowitz 2017). Similar to the Self-Reliant Rover approach, the use of onboard scheduling is intended to allow the Mars 2020 rover to use current vehicle knowledge when generating schedules to accomplish mission objectives. This will reduce the loss of productivity that results from the difficulty in predicting how much resources (e.g. time and energy) activities will consume. The Self-Reliant Rover approach is addressing additional productivity challenges by improving the ability of rovers to identify their own objectives, to incorporate a richer set of guidance from operators and to reason about slip hazards as it navigates.

Conclusion

We have demonstrated an anytime planning approach that incorporates operator intent and is capable of reacting to changes encountered during execution. In isolated tests as well as a realistic integrated system evaluation, these capabilities were shown to provide a measurable improvement in over all productivity.

Acknowledgements

This research was conducted at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work was funded by the Jet Propulsion Laboratory Research and Technology Development program.

References

Boddy, M. 1991. Anytime problem solving using dynamic programming. In *Proceedings of the ninth National conference on Artificial intelligence-Volume 2*, 738–743. AAAI Press.

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

Chattopadhyay, D.; Mishkin, A.; Allbaugh, A.; Cox, Z. N.; Lee, S. W.; Tan-Wang, G.; and Pyrzak, G. 2014. The Mars Science Laboratory supratactical process. In *Proceedings of the SpaceOps 2014 Conference*.

Chien, S.; Rabideau, G.; Knight, R.; Sherwood, R.; Engelhardt, B.; Mutz, D.; Estlin, T.; Smith, B.; Fisher, F.; Barrett, T.; Stebbins, G.; and Tran, D. 2000. Aspen - automating space mission operations using automated planning and scheduling. In *International Conference on Space Operations (SpaceOps 2000)*.

Chien, S.; Rabideau, G.; Tran, D.; Doubleday, J.; Nespoli, F.; Ayucar, M.; Sitje, M.; Vallat, C.; Geiger, B.; Altobelli, N.; Fernandez, M.; Vallejo, F.; Andres, R.; and Kueppers, M. 2015. Activity-based scheduling of science campaigns for the rosetta orbiter. In *Proceedings of IJCAI 2015*.

Francis, R.; Estlin, T.; Doran, G.; Johnstone, S.; Gaines, D.; Verma, V.; Burl, M.; Frydenvang, J.; Montano, S.; Wiens, R.; Schaffer, S.; Gasnault, O.; DeFlores, L.; Blaney, D.; and Bornstein, B. 2017. AEGIS autonomous targeting for ChemCam on Mars Science Laboratory: Deployment and results of initial science team use. *Science Robotics* 2(7).

Gaines, D.; Doran, G.; Justice, H.; Rabideau, G.; Schaffer, S.; Verma, V.; Wagstaff, K.; Vasavada, A.; Huffman, W.; Anderson, R.; Mackey, R.; and Estlin, T. 2016. Productivity challenges for Mars rover operations: A case study of Mars Science Laboratory operations. Technical Report D-97908, Jet Propulsion Laboratory.

Gaines, D.; Russino, J.; Doran, G.; Mackey, R.; Paton, M.; Rothrock, B.; Schaffer, S.; akbar Agha-mohammadi, A.; Joswig, C.; Justice, H.; Kolcio, K.; Sawoniewicz, J.; Wong, V.; Yu, K.; Rabideau, G.; Anderson, R.; and Vasavada, A. 2018. Self-reliant rover design for increasing mission productivity. In *2018 ICAPS Workshop on Planning and Robotics (PlanRob 2018)*. Also appears at International Symposium on Artificial Intelligence, Robotics, and Automation in Space (ISAIRAS 2018).

Klenk, M.; Molineaux, M.; and Aha, D. W. 2013. Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence* 29:187–206.

Mali, A. D. 2016. Expressing user intent in planning by instance rewriting. In *Proceedings of the 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI 2016)*.

Paar, G.; Woods, M.; Gimkiewicz, C.; Labrosse, F.; Medina, A.; Tyler, L.; Barnes, D. P.; Fritz, G.; and Kapellos, K. 2012. PRoViScout: a planetary scouting rover demonstrator. In *Proceedings of SPIE Vol. 8301 Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*.

Rabideau, G., and Benowitz, E. 2017. Prototyping an onboard scheduler for the Mars 2020 rover. In *Proceedings of the International Workshop on Planning and Scheduling for Space*.

Shalin, V. L.; Wales, R. C.; and Bass, D. S. 2005. Communicating intent for planning and scheduling tasks. In *Proceedings of HCI International*.

Wettergreen, D.; Foil, G.; Furlong, P. M.; and Thompson, D. R. 2014. Science autonomy for rover subsurface exploration of the atacama desert. *AI Magazine* 35(4).

Woods, M.; Shaw, A.; Barnes, D.; Price, D.; Long, D.; and Pullan, D. 2009. Autonomous science for an ExoMars rover-like mission. *Journal of Field Robotics* 26(4):358–390.

Yingst, R.; Berger, J.; Cohen, B.; Hynek, B.; and Schmidt, M. 2017. Determining best practices in reconnoitering sites for habitability potential on Mars using a semi-autonomous rover: A geoheuristic operational strategies test. *Acta Astronautica* 132.