

# Scalable information-theoretic path planning for a rover-helicopter team in uncertain environments

*International Journal of Advanced Robotic Systems*  
March-April 2021: 1–18  
© The Author(s) 2021  
Article reuse guidelines:  
[sagepub.com/journals-permissions](http://sagepub.com/journals-permissions)  
DOI: 10.1177/1729881421999587  
[journals.sagepub.com/home/axr](http://journals.sagepub.com/home/axr)



**Larkin Folsom<sup>1</sup>, Masahiro Ono<sup>2</sup>, Kyohei Otsu<sup>2</sup>  
and Hyoshin Park<sup>1</sup> **

## Abstract

Mission-critical exploration of uncertain environments requires reliable and robust mechanisms for achieving information gain. Typical measures of information gain such as Shannon entropy and KL divergence are unable to distinguish between different bimodal probability distributions or introduce bias toward one mode of a bimodal probability distribution. The use of a standard deviation (SD) metric reduces bias while retaining the ability to distinguish between higher and lower risk distributions. Areas of high SD can be safely explored through observation with an autonomous Mars Helicopter allowing safer and faster path plans for ground-based rovers. First, this study presents a single-agent information-theoretic utility-based path planning method for a highly correlated uncertain environment. Then, an information-theoretic two-stage multiagent rapidly exploring random tree framework is presented, which guides Mars helicopter through regions of high SD to reduce uncertainty for the rover. In a Monte Carlo simulation, we compare our information-theoretic framework with a rover-only approach and a naive approach, in which the helicopter scouts ahead of the rover along its planned path. Finally, the model is demonstrated in a case study on the Jezero region of Mars. Results show that the information-theoretic helicopter improves the travel time for the rover on average when compared with the rover alone or with the helicopter scouting ahead along the rover's initially planned route.

## Keywords

Path planning and navigation, mobile robots and multirobot systems, multiagent robot teams, sensor and information fusion, robot sensors and sensor networks, uncertainty reduction, machine vision, vision systems

Date received: 6 November 2020; accepted: 31 January 2021

Topic Area: Mobile Robots and Multi-Robot Systems

Topic Editor: Lino Marques

Associate Editor: Francesco Amigoni

## Introduction

In highly uncertain environments, maximizing information gain is vital to ensuring safe, efficient, autonomous exploration. This research enhances the scientific and engineering value of autonomous vehicles by finding the fastest traversable routes in uncertain environments. The information of the highest value in an uncertain environment is found in regions in which the rover's speed probability distribution has high standard deviation (SD). In other

<sup>1</sup>Department of Computational Science and Engineering, North Carolina A&T State University, Greensboro, NC, USA

<sup>2</sup>Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

### Corresponding author:

Hyoshin Park, Department of Computational Science and Engineering, North Carolina A&T State University, Greensboro, NC, USA.

Email: [hpark1@ncat.edu](mailto:hpark1@ncat.edu)

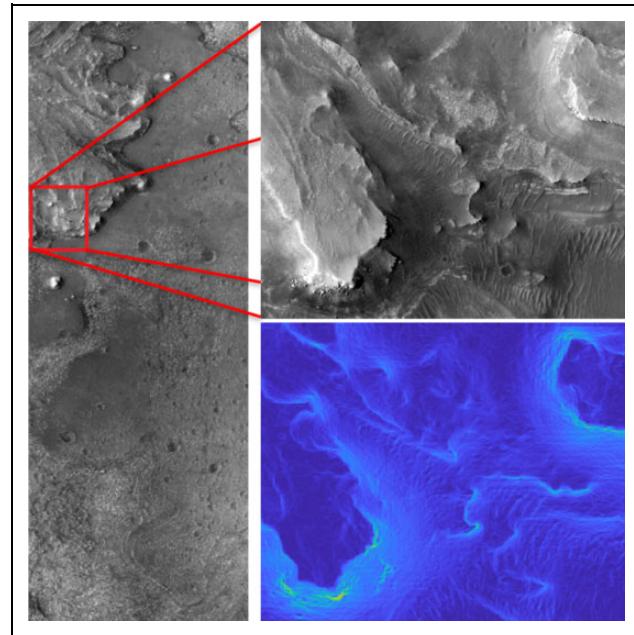


words, the most important information is found in the regions in which true speed of the rover could deviate significantly from the expected value. Efficient exploration of these regions with an unmanned information-theoretic helicopter reduces the rover's travel time uncertainty and enables the rover to adjust its planned route if the conditions inside the highly uncertain region are found to be more optimal.

Traditional uncertainty metrics such as Shannon entropy do not distinguish between different bimodal distributions by treating all information as equally valuable.<sup>1</sup> Nonmetric measures of information gain, such as Kullback–Leibler (KL) divergence, provide an appealing alternative.<sup>2</sup> KL divergence has been used in machine learning-based image processing and has been shown to work for unimodal distributions.<sup>3</sup> However, KL divergence presents a problem for multimodal distributions by introducing bias toward only one mode (e.g. exclusive, reverse KL divergence) or toward the mean of the modes (e.g. inclusive, forward KL divergence). The data set for this study contains non-Gaussian bimodal probability distributions and representing the information gain using KL divergence requires comparison to an ideal distribution. This biases the agent toward exploring only some types of uncertain regions while ignoring other potentially valuable uncertain regions. Therefore, uncertainty in expected rover travel time is represented as the SD of the rover's travel time probability distribution.

In this study, two types of path-planning models are presented. First, a utility-based information-theoretic model for routing a single agent through an uncertain correlated grid environment is presented. Then, a rapidly exploring random tree (RRT)-based two-stage multiagent path-planning algorithm is presented, which computes travel time optimized and safe routes for a Mars rover to successfully navigate through rugged, uncertain terrain with the aid of a cooperating information-theoretic helicopter. In the proposed RRT-based path-planning algorithm, knowledge about the environment is gained by the helicopter and transmitted to the rover allowing the rover to optimize its path. The rover travel time probability distribution for the surface is generated based on a rover mobility model (RMM).<sup>4</sup> Inputs to the RMM are the terrain type determined using the soil property and object classification (SPOC) terrain classifier, slope data from a digital elevation model (DEM) generated using HiRISE stereo images (Figure 1), and rock abundance in terms of the cumulative fractional area (CFA) covered by rocks.<sup>5,6,7</sup>

The information-theoretic helicopter is routed using an information-theoretic RRT algorithm (**RRT\*-IT**), which balances the cost of travel with the reward of sequential information gain. The rover's RRT algorithm (**RRT\*-ETT**) considers the expected travel time (ETT) while avoiding uncertain regions with high SD. RRT algorithms with edge cost based on ETT and the ability to be rewired (e.g. RRT\*) provide computationally fast path

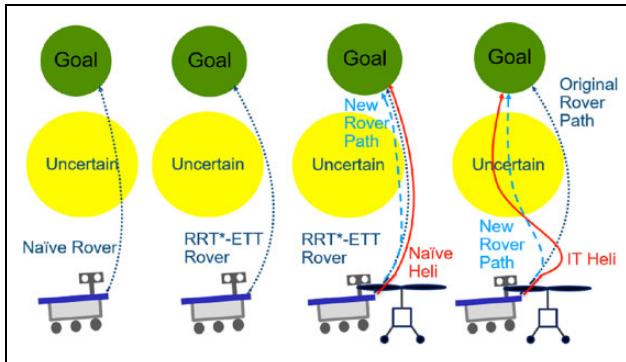


**Figure 1.** Left: HiRISE image of Jezero region on Mars. Top right: Detailed view of ROI. Bottom right: Slope map for ROI generated from stereo pair digital elevation model. ROI: region of interest.

planning and replanning on a large-scale, high-resolution environment.

Previous studies have focused on minimizing risk to the rover by not visiting highly uncertain, low-confidence regions, but the potential exists for these regions to be traversable. Scouting these regions may offer significant travel time savings under certain conditions; however, naively entering a region of high uncertainty without scouting is a high-risk/high-reward strategy that increases the risk of getting stuck. For this study, the Mars environment is assumed to be static over the time scales involved, but knowledge about the environment is dynamic. We compare the results for four Mars scenarios: rover alone using naive shortest path planning, rover alone using safe path planning, rover using safe path planning with helicopter scouting the planned rover path, and rover using safe path planning with information-theoretic helicopter (Figure 2).

The presented rover + information-theoretic helicopter methodology is best suited to cases in which imperfect a priori knowledge about an unexplored environment is available, the cooperating helicopter's search cannot be exhaustive, there is a need to reduce travel time where possible, the penalty for failure is high, and computational hardware is limited. The presented methodology is not well suited to scenarios in which a rover-helicopter team must cooperate to traverse a completely unexplored environment or for scenarios in which travel time is not a concern and safety can be completely prioritized. Because of the specific nature of this problem in which reasonably accurate a priori knowledge of the environment exists, the median travel time reduction seen with the addition of an



**Figure 2.** 1: Rover alone using standard RRT\*. 2: Rover alone using RRT\*-ETT. 3: Rover using RRT\*-ETT with helicopter scouting planned route. 4: Rover using RRT\*-ETT with helicopter searching nearby uncertain regions using RRT\*-IT to find better routes. RRT: rapidly exploring random tree; ETT: expected travel time; RRT\*-IT: rapidly exploring random tree-information theoretic.

information-theoretic helicopter is not always expected to be extreme.

For example, if a map's terrain classifications are known with reasonably high certainty, a path, which always assumes that the maximum probability classification is true, can perform acceptably in a majority of cases. The problem with this assumption is most apparent in terms of the worst-case performance, as some fraction of trips will end with the rover becoming stuck. For the Mars rover case, getting stuck means the failure of an extremely expensive and time-consuming mission. Alternatively, path planning using the worst-case travel time for each grid cell in the map can provide maximum safety but results in an overly cautious route, which is excessively slow in many cases and inappropriate for missions with time restrictions or deployments in which saved time allows the rover to perform additional nondriving-related scientific tasks.

## Related work

The presented scalable autonomous path-planning algorithm will determine routes that optimize travel time for a rover on the surface of Mars. In robotic path planning, risk is often assumed to be a binary parameter in which a region is either obstructed or clear, and the path costs are assumed to be a deterministic function of Euclidean distance. Assuming stochastic path costs can present significant scalability problems, which prohibit real-time path planning in a full-scale environment. In the proposed research, terrain properties are probabilistic and traversability of the environment is not certain for the rover. The addition of a helicopter enables exploration of the most uncertain regions without risk for the rover.

Prior studies on the topic of scalable path planning for a Mars rover used a variant of Dijkstra's algorithm to generate a path.<sup>4,8</sup> Their work assumed the existence of hard

obstacles (e.g. cells are either obstructed or clear), used prespecified regions of interest, and considered only the rover itself without a cooperating helicopter. The use of a cooperating helicopter and a Markov decision process (MDP) to find optimal paths has been explored; however, using MDPs to solve the path planning problem introduces additional complexity due to time and path-dependent rewards for information gain.<sup>9</sup> The problem does not strictly conform to the Markov property due to rewards being based on the previous spatiotemporal locations of the agent; therefore, solving the problem using MDPs requires methods, which can degrade computational performance or the quality of the result. A similar Mars rover navigation problem was solved using an MDP formulation; however, this research only considered slope when evaluating traversability, assumed a coarse  $20 \times 20 \text{ m}^2$  cell size, and did not consider a cooperating helicopter.<sup>10</sup> The addition of terrain type and rock abundance to their model could significantly increase the state and observation spaces.

Prior research has explored the use of a cooperating rover and drone for a small-scale rover navigation problem.<sup>11</sup> The authors compared a D\* algorithm with their response time algorithm for rover path planning and did consider speed and traversability for different terrain types. Two drone exploration strategies were considered: greedy search and exhaustive search. Only three terrain types were considered: concrete, water, and grass, with water serving as a hard obstacle. The authors did not consider a priori probabilistic terrain classifications to inform their search, “no map, and no information about the terrain classes, is available a priori.”<sup>11</sup> Additionally, the authors assumed that the highest probability classification from their convolutional neural network classifier is the true classification.<sup>11</sup>

A safe ship navigation problem with the goal of avoiding sea mines was solved under three different strategies, including the shortest path, least maneuvers, and a combined strategy.<sup>12</sup> Their developed algorithm was based on the A\* algorithm and used a large map size of  $6 \times 4.5 \text{ km}^2$ . The safe ship navigation problem was solved in 2D space with mines serving as hard obstacles and did not consider speed or traversability, only turning radius constraints.<sup>12</sup> Solution time varied depending on the level of optimality desired. Because their algorithm is online, the solution occurs in two stages with preprocessing times ranging from 1.3 min to 35.7 min and planning times ranging from 0.8 s to 16.2 s.

Performance comparisons of path constrained safe interval path planning and any-angle safe interval path planning algorithms for solving a rover path-planning problem with dynamic obstacles have been explored.<sup>13</sup> Dynamic obstacle trajectories were assumed to be known a priori and the authors did not consider a cooperating helicopter or a traversability model. Their map size was  $46 \times 70 \text{ m}^2$  with  $1 \text{ m}^2$  grid cells, and no solution times were included in the results.

Safe robot navigation in 2D and 3D spaces formulated as a mixed observability MDP problem and solved using a variant of the Monte-Carlo tree search (MCTS) algorithm.<sup>14,15</sup> Their research produced reasonable but suboptimal path plans on two simple maps. The authors considered only the presence of hard obstacles in their maps without a traversability model and did not consider a cooperating helicopter. The authors did not include solution times in their results aside from setting the maximum runtime to 180 min.

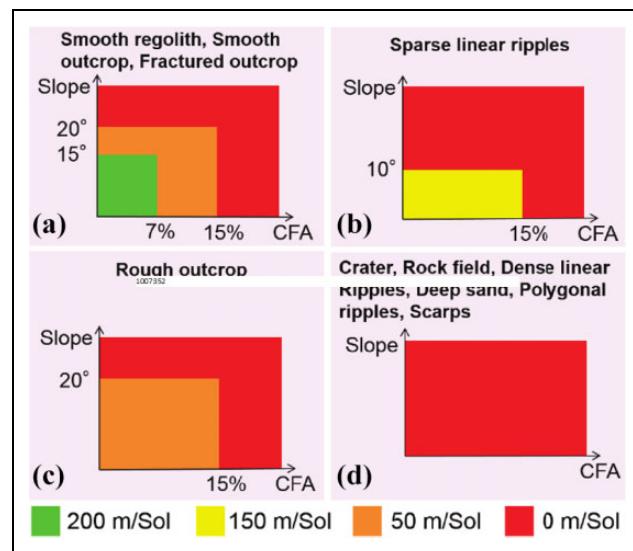
Traditional partially observable MDP (POMDP) approaches are less computationally tractable when compared with MDPs.<sup>16</sup> While simple 2D POMDP models for aircraft and autonomous ground navigation research have been successful, these studies lack the high 1 m<sup>2</sup> resolution required for safe scalable autonomous navigation on an uncertain Mars environment.<sup>17,18</sup> The use of traditional POMDP approaches for this problem would require coarse sampling of the region or solutions for very small regions. Our research focuses on regions containing 40,000 to 673,854 1 m<sup>2</sup> cells allowing for multiple days (Martian Sol) of rover travel time and is beyond the scope of traditional POMDP approaches.

Scalable algorithms for eventual deployment on autonomous Mars-based vehicles will make use of onboard computing. This requirement rules out algorithms, which rely on Earth-based computers leveraging both CPU and GPU parallelization, as well as high power consumption, to achieve scalability. While research into highly parallelizable POMDPs such as DESPOT- $\alpha$  is promising, these algorithms perform well because of massively parallelizable hardware found in modern standalone GPUs.<sup>19</sup> Such hardware is unavailable for space-faring rovers, which continue to rely on simpler single-board computers with large radiation-resistant transistors, such as the BAE Systems RAD750.<sup>20</sup>

Guided RRT algorithms (e.g. solving RRTs for potential fields) offer efficient solutions in stochastic environments.<sup>21</sup> RRT algorithms are also effective for path planning in dynamic environments and for multiple agents through node sharing.<sup>22</sup> In addition, RRT algorithms can benefit from parallelization, further improving performance and scalability.<sup>23</sup>

The use of a multiagent information-sharing communication approach for solving path planning, routing, and scheduling problems has shown to be successful in recent years.<sup>24,25,26</sup> Multiagent methods have been used previously for tasks such as multirobot exploration and search-and-rescue.<sup>27</sup> Individualized metaheuristic/local search combinations have also been used to schedule and route multiple agents.<sup>25</sup> Multiagent approaches have also been used to effectively plan optimal taxiing routes for aircraft with three types of agents: resource management, aircraft, and resource node agent.<sup>28</sup>

In the transportation research above, routing uncertainty typically addresses variance in traffic patterns through the



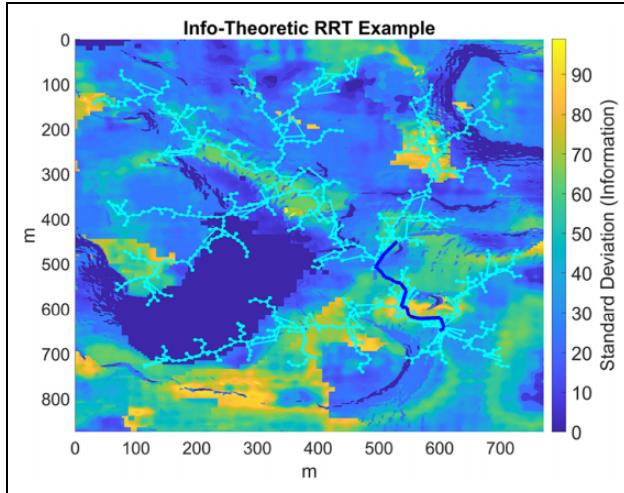
**Figure 3.** Rover mobility model.<sup>4</sup> (a) Smooth regolith, smooth outcrop, and fractured outcrop; (b) sparse linear ripples; (c) rough outcrop; and (d) crater, rock field, dense linear ripples, deep sand, polygonal ripples, and scars.

day or variability in bus transfer times. There are many studies on human decision making based on travel time reliability, but prediction of travel time and link speed still depend on traffic sensors and other infrastructure.<sup>29,30,31,32,33</sup> Travel time prediction therefore depends on sample data from the network. If the sample data is inaccurate, resulting in unexpected low travel speed or nontraversability, significant costs are incurred by the user. To minimize the disadvantage of uncertain classification and link travel times, this research proposes an anticipatory methodology for a helicopter to visit select regions based on the terrain type probability distribution. The cost of visiting a region is balanced by the benefit of gaining information about that region.

### Contribution

This work introduces new methods for a multiagent cooperative robot-helicopter team to perform safe scalable path planning in uncertain environments using RRTs. RRT algorithms are sampling-based approaches, which enable computationally efficient exploration of large environments. Our information-theoretic helicopter explores regions with high uncertainty, eliminating risk for the mission-critical rover while providing improved travel time on average and greater safety for the rover.

We introduce two extensions to the RRT\* algorithm, which generate paths based on cost functions. The **RRT\*-ETT** algorithm generates a path plan for the rover by calculating route cost as ETT, which is calculated as the path integral of the inverse expected speed based on the RMM shown in Figure 3 along the 3D surface. The **RRT\*-IT** algorithm balances the cost of travel time with



**Figure 4.** Example run of **RRT\*-IT** code on the Jezero region of Mars, showing the tree growing along routes of greatest utility in terms of the travel cost and information gain for the Mars helicopter. **RRT\*-IT**: rapidly exploring random tree-information theoretic.

the reward of information gain to find the best locations for the Mars helicopter to observe. Figure 4 shows a partial construction of the **RRT\*-IT** tree, which considers information gain as the reduction in the SD of the probability distribution of expected rover speed at each location.

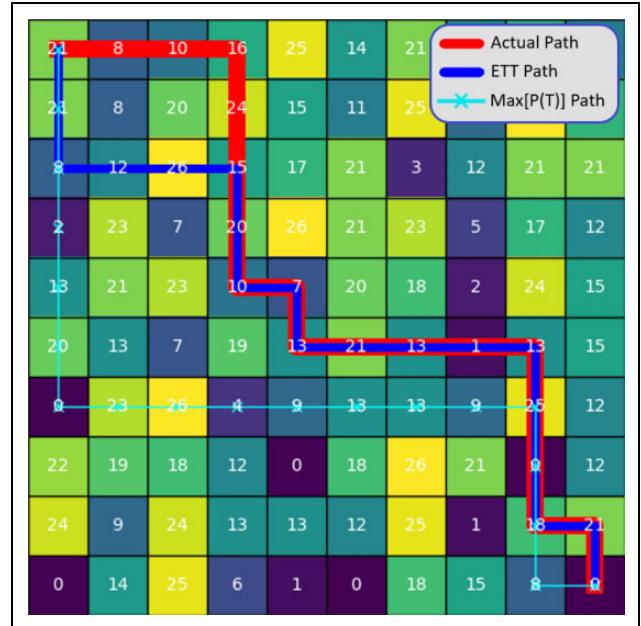
## Information-theoretic path planning

This section presents a utility-based methodology for improving the travel time of a single agent in an uncertain environment by balancing exploring for information gain and exploiting the information through travel time reduction. The environment for this scenario, shown in Figure 5, is a small  $M \times M$  grid with uncertain travel time through each cell. In this scenario, observations of one cell provide information about unobserved cells of the same type.

### Single-agent information-theoretic scenario

A detailed description of a utility-based single-agent path planning scenario is presented in this subsection, while subsequent sections demonstrate a methodology for applying this utility-based model to an RRT\* framework with single and multiple agents. Assume an  $M \times M$  grid with binary travel time probability distribution for each cell, a single agent ( $i = 1$ ), and no time-dependence of the grid. For small grids, it is possible to compute all feasible paths using depth first search. For large grids, a stochastic method such as random search or a sampling-based approach such as RRTs may be used. In this section, the utility-based methodology is compared to two other path-planning methods.

For each path, three travel times are created: the best-case travel time, worst-case travel time, and ETT. The



**Figure 5.** Single agent: Three paths compared on a grid map with entropy minimization. The actual path (utility function optimized) is the path based on the utility function in 9, while the other two are theoretical paths, which minimize the initial expected travel time or the  $\text{Max } [P(T)]$  (highest probability classification)-based travel time. The grid shows each cell type by number (white) and filled color. Note that the highest information gain is achieved by visiting the maximum number of different cell types.

worst-case travel time assumes that the travel time for a given path is the sum of the maximum travel time through each cell. The best-case travel time assumes that the travel time for a given path is the sum of the minimum travel time through each cell  $j$ . The  $\text{ETT}_{i,p}$  for agent  $i$  on path  $p \ni J$  containing  $J$  cells is defined in equation (1)

$$\text{ETT}_{i,p} = \sum_{j=1}^J \sum_{y=1}^Y [P_{j,y} T_y] \quad (1)$$

where  $P_{j,y}$  is the probability of the cell  $j$  being type  $y$ ,  $T_y$  is the travel time through a cell of type  $y$ , and  $Y$  is the total number of cell types.

**Defining and calculating entropy.** Due to uncertainty in the environment, the agent should explore to gather information rather than naively following the route, which is expected to give the fastest travel time. This incentive to explore and gather information is specified by a utility function in equation (2). Balancing the trade-off between reward (information gain) and cost (increased travel time) determines which route to take. Two weighting variables,  $W_{i,1}$  and  $W_{i,2}$ , are used to alter agent  $i$ 's exploration preferences and approximate the normalization of the cost and reward variables, which are in different units. In the RRT\* sections of this article, we also find that traditional normalization techniques are not feasible, because each edge of

the tree has only a single travel time and single information gain value. Additionally, normalization fails in the case of using the range of possible travel times (0 to infinity) for normalization. Therefore, we approximate normalization using the user-defined weighting variables  $W_{i,1}$  and  $W_{i,2}$

$$U_{i,p,t} = W_{1,i} \text{reward}_{i,p,t} - W_{2,i} \text{cost}_{i,p,t} \quad (2)$$

The reward component in equation (2) is defined in terms of the information gained by visiting that cell. Consider an environment which contains only three possible terrain types: type 1, type 2, and type 3. Initial satellite observation indicates that the cell (3,5) is classified as terrain type 1. If an agent visits cell (3,5) and finds it to be of terrain type 2, then it has new information regarding every other cell initially classified as type 1. For this simplified model, the optimal route in terms of information gain would be the route, which gained the most information in the shortest number of steps. The most optimal path not only considers how much information is gained but also considers how quickly the information is gathered. Many paths gather a large amount of information, but fewer paths can gather information quickly.

In a highly correlated grid, not all cell exploration is equally valuable because some cells contain more information than other cells. This is due to each cell's potential for variance from the ETT and its correlation with other cells in the grid. For example, if there are an equal number of cells of each terrain type in the environment, then the most valuable cells would be the cells in which the variance of ETT is highest. These cells have the greatest potential to change the travel time for the agent. Knowledge of the true state of cells with large travel time SD is more valuable when selecting a route than knowing the true states of cells with a small travel time variance because knowing the latter cells' true states will not cause the ETT to deviate as significantly from the original prediction. In the binary travel time grid-based scenario, we represent this variance as the difference between the maximum and minimum travel times  $T_2 - T_1$  within each cell.

If the number of cells of each terrain type is different, then the abundance of cells of each type must also be considered when determining which cells contain the most information in a correlated grid environment. If there is only one cell of a given type in the entire grid, then knowing its state does not provide as much information gain as knowing the state of a cell, which has four other identical cells in the grid. Equation (3) accounts for the number of cells of a given type. This equation comes from the concept of informational entropy and provides a measure of the information gained by visiting a cell whose terrain type is specified by the variable  $y$  under the assumption of perfect observation

$$\text{Reward} = \frac{n_y}{N} \quad (3)$$

where  $n_y$  is the number of cells of type  $y$  and  $N$  is the total number of cells in the grid.

The cost component in equation (2) is computed by comparing the difference in ETTs between the current route and the route with the fastest ETT, as shown in equation (4). If there is a large increase in travel time for the current route compared with the fastest ETT route, then the cost increases. Equation (4) becomes negative when the fastest ETT route has a higher travel time than the detour route. By subtracting a negative value, the utility function is increased and this type of route is given more incentive. If the detour takes longer than the fastest ETT route, then the utility is reduced

$$\text{Cost}_{i,p,t} = \text{ETT}_{i,\text{detour},t} - \text{ETT}_{i,\text{fastest},t} \quad (4)$$

where  $\text{ETT}_{i,\text{fastest},t}$  is the fastest ETT and  $\text{ETT}_{i,\text{detour},t}$  is the travel time of the path for which the utility function is being calculated. The extent of this increase or decrease is controlled by the weighting variable  $W_{2,i}$  in equation (5) on the difference in travel times for the fastest path and the detour path. This variable serves to prevent cost from scaling with grid size, since the reward does not

$$W_{2,i} = W_{\text{cost},i} = \frac{1}{N} \quad (5)$$

To disincentivize collecting rewards late in the path, a discount factor  $\lambda^k$  is used, where  $k$  is the number of discrete steps taken along the path. A typical value used in this model is  $\lambda = 0.98$ ; however, this value can be adjusted to achieve the desired amount of exploration, with lower  $\lambda$  values making exploration less rewarding to the agent. The cost for taking a path is defined by the additional ETT incurred by taking that path versus the path with the fastest ETT. The utility function for agent  $i$  is now defined by equation (6), where  $p$  is a unique path

$$U_{i,p,t} = \text{reward}_{i,p,t} - W_{2,i} \text{cost}_{i,p,t} \quad (6)$$

Equation (7) is the path reward

$$\text{Reward}_{i,p,t} = \sum_{j=0}^{\text{End}} \left[ \frac{W_{1,j} \lambda^k n_{y,j,t} (T_{\max,j,t} - T_{\min,j,t})}{N} \right] \quad (7)$$

and equation (8) is the weighting function

$$W_{1,j} = \log_2 \left[ \frac{1}{\text{Max}[P_{j,t}(T_{\min,j,t}), 1 - P_{j,t}(T_{\min,j,t})]} \right] \quad (8)$$

where  $j$  is a cell in the path,  $n_y$  is the number of cells of a given terrain type,  $N$  is the total number of cells,  $T_{\max,j,t}$  is the greater of the two times in the binary travel time cell at time  $t$ , and  $T_{\min,j,t}$  is the lesser of the two times in the binary travel time cell.

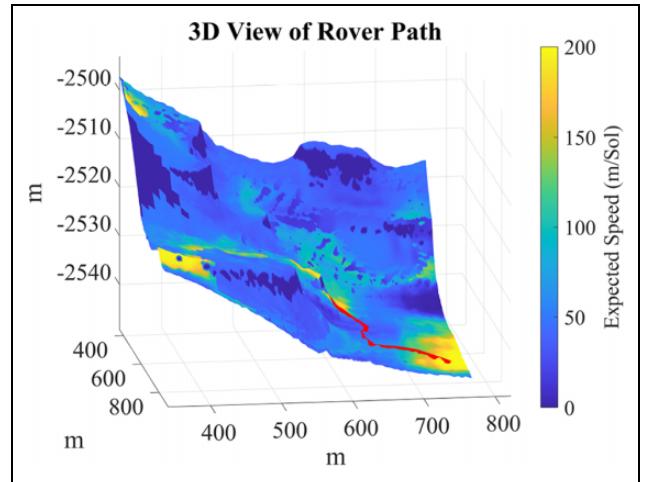
A probability-based calculation will further improve the likelihood of choosing the best route, and therefore, the reward should take the probability distribution into account. The probability component of the utility function,

shown in equation (8), is taken from the representation of probability distributions in terms of their entropy.

**Consideration of probability distribution accuracy.** In real-world applications of this model, discrepancies may exist between the predictions made using sensor analysis and the actual conditions on the ground. The extent of these discrepancies is not known beforehand, and therefore, the accuracy of the probability distribution should be defined. The accuracy of the probability distribution is represented by an error term in the utility function. The error is reduced during the trip any time a cell of the same type is visited and the state is confirmed to be identical to the original cell. Because the error is not assumed to be identical for all cells, it can only be improved when visiting a cell of the same type after visiting a cell of that type initially. A single sample from a probability distribution cannot confirm the shape of the distribution. To reduce the error, resampling cells with the same probability distribution is performed.

This technique has the added benefit of correcting the correlation assumption we use. If cells with the same probability distribution are 100% correlated, the error will decrease as the trip progresses. An error reduction will produce greater confidence in the chosen path, ensuring safer routing and more accurate travel time prediction. If a situation is encountered in which the correlation assumption does not hold, the error will increase as successive cells of the same type are visited and found to be in different states. This will have the effect of weakening the correlation assumption in the algorithm. Our algorithm has the capability of adapting to real-world information about the state of the grid, appropriately adjusting the predicted state of unexplored grid cells based on information obtained during the trip. It is also possible that correlation assumption is valid only in certain regions of the grid. In this case, the error term will allow the agent to adapt to the changes in its environment. The error can increase or reduce the ETT. Our model will add negative error from the maximum  $P(t)$ , providing a worst-case uncertainty with regard to the state of the cell when path planning.

Once a cell of a given type is visited, the state of all cells of the same type are assumed to be known within some error. This error term,  $\varepsilon$ , is reduced as subsequent correlated cells are visited and found to be in the same state as the first cell of that type. The error grows if subsequent correlated cells are visited and found to be in a different state than the first cell of that type. The agent does not directly receive a reward for visiting a cell, which is identical to one which it has already visited. However, updating the error term provides an indirect reward to visiting cells of the same type. The error term is included in the probability component of equation (9) as an addition to the probability component of the reward weighting function. The reward is increased when the cells are found to have



**Figure 6.** Example 3D view of a path on the Mars terrain simulation region (same region as Figure 7).

been correlated and is decreased when the cells are found to not be correlated

$$W_{j,1} = \log_2 \left[ \frac{1}{\text{Max}[P_{j,t}(T_{j,1,t}), 1 - P_{j,t}(T_{j,1,t})] + \varepsilon} \right] \quad (9)$$

**Consideration of trade-off.** Routing an agent through a grid that contains uncertainty requires a method to measure which routes are most likely to be effective. The objective of the agent dictates the effectiveness of a given route. In some cases, the objective is to explore the grid to minimize the uncertainty. In other cases, the objective is to move through the grid as quickly as possible.

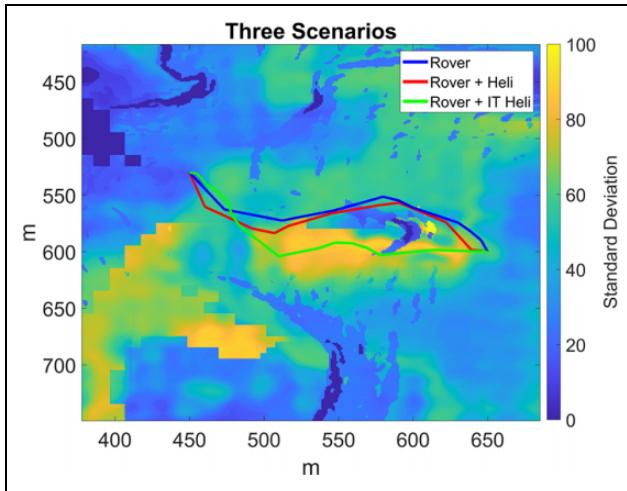
When the uncertainty in a grid is a simple binary probability, such as a grid whose cells are either blocked or clear, the entropy of an individual cell can be described using Shannon entropy in equation (10). The probability that a cell is clear is given by  $p(c)$  and the probability that the cell is blocked is  $(1 - p(c))$

$$H_C(c) = -p(c)\log_2[p_c] - (1 - p(c))\log_2[1 - p(c)] \quad (10)$$

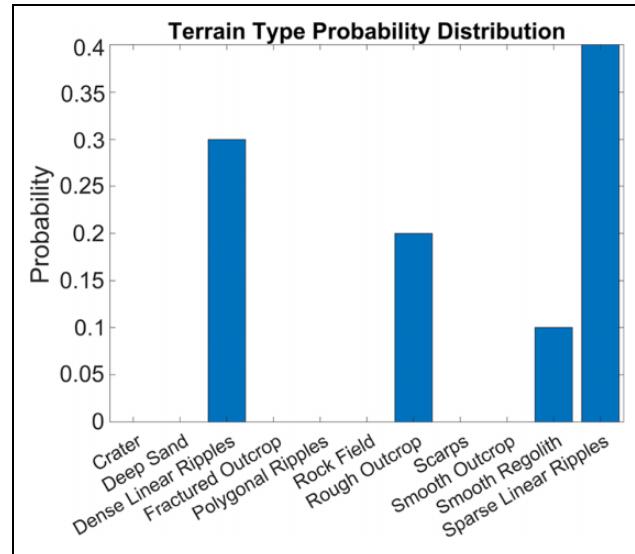
## Mars terrain and expected speed models

The RRT-based two-stage multiagent path-planning algorithm uses the 3D Mars terrain model (Figure 6) and rover expected speed model presented in this section. Path planning for the rover is accomplished using an ETT-based extension of the RRT\* algorithm called **RRT\*-ETT**. We define the environment as a discretized grid with each grid cell representing one pixel from the 3D stereo satellite imagery (Figure 6). The use of a grid structure allows the model to be applied to images of any region. Map resolution is 1 m<sup>2</sup> per pixel and each cell in the grid is 1 m<sup>2</sup>.

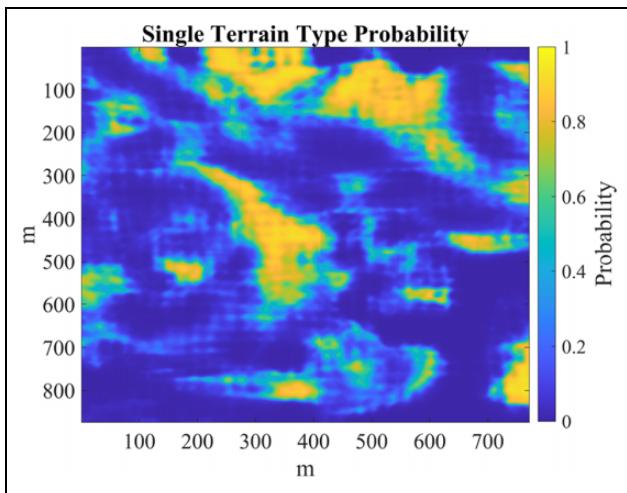
The initial state of each grid cell is defined by four variables obtained from satellite data, with the fourth



**Figure 7.** Case study on Jezero region showing path plans for three scenarios. Map color represents expected speed probability distribution standard deviation. True travel time for **rover alone** is 2.62 Sol. True travel time for **rover + naive helicopter** is 2.55 Sol. True travel time for the proposed **rover + IT heli** model is 1.68 Sol.



**Figure 9.** Example terrain type probability distribution.



**Figure 8.** Probability map for one of the 11 terrain classes, smooth regolith, in Jezero region.

variable being used to calculate the distance between any two points. These are the slope, the terrain type probability distribution in Figure 8, the CFA of rocks, and the elevation. Slope data is calculated using a DEM generated from stereo pair HiRISE images using the Geospatial Data Abstraction Library.<sup>34</sup> CFA data are obtained using the methodology developed by Golombek et al.<sup>7</sup>

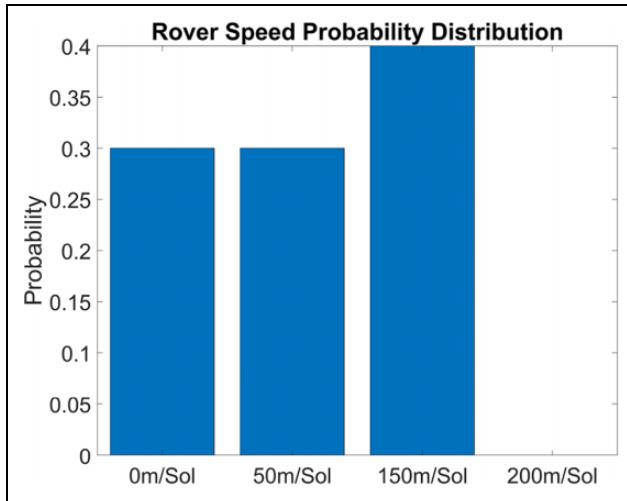
There are 11 possible terrain types, which are classified using the SPOC algorithm.<sup>5</sup> SPOC generates a probability distribution over the 11 terrain type classes. Because SPOC is a machine learning algorithm working with image data, most grid cells have some uncertainty in their terrain type classification.

From the first three variables defining the cell state, an expected speed probability distribution is generated for each cell. This distribution is consistent with prior work and contains four discrete speed possibilities for any combination of the three state variables.<sup>4</sup> The possible speed classifications are 0, 50, 150, and 200 m/Sol.

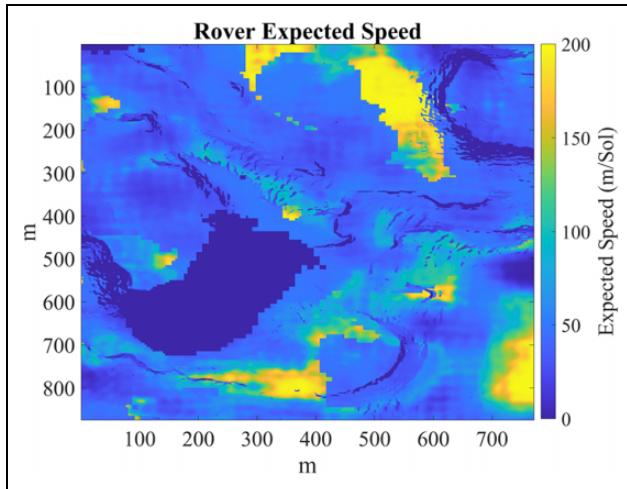
#### Example expected speed calculation

Generation of expected speed probability distribution is accomplished using an RMM with three input variables: slope, rock abundance as a CFA, and terrain type. The slope is calculated using a DEM from HiRISE satellite data and the CFA is obtained using the methodology developed by Golombek et al.<sup>7</sup> Next, the terrain type is determined using a deep learning-based image classification algorithm, SPOC, which generates the 11 class discrete terrain type probability distribution for each pixel of the HiRISE image.

There are four possible speed outputs for the RMM based on the slope, CFA, and terrain type. For example, if a pixel in the HiRISE image has a slope of 5°, CFA of 10%, and terrain type of smooth regolith, the expected rover speed based on the RMM is 50 m/Sol. SPOC does not always provide certainty with the terrain type classification. Many pixels will have a terrain type probability distribution. For pixels with uncertain terrain type classification, the terrain type probability distribution from the SPOC algorithm is used to generate an expected speed probability distribution from the RMM. Given a pixel with the terrain type probability distribution shown in Figure 9, a slope of 5°, and CFA of 10%, the resulting expected speed probability distribution is shown in Figure 10. This calculation was performed for



**Figure 10.** Example rover speed probability distribution.



**Figure 11.** Expected rover speed based on slope, CFA, and terrain type probability distribution.

each pixel on a subregion of the Jezero region on Mars. The results are shown in Figure 11.

## Multiagent scenarios

### RRT\*-Expected travel time

Path planning for the rover is accomplished using an extension of the RRT\* algorithm called **RRT\*-ETT**. This algorithm functions as a standard RRT\* algorithm with the exception that the maximum expected edge travel time is used in place of the maximum edge length.<sup>35,36</sup> Because the algorithm is applied to a 3D surface with variable travel time, the ETT along each branch in the tree is computed by taking the line integral of the pace along the branch. This algorithm is robust to information error because it always

assumes the worst case. When the tree is rewired during each iteration of tree growth, we again use the ETT for the length of the rewired edge. The advantage of using RRT\* in a large-scale environment is that it converges on reasonable solutions without an excessive number of iterations and without considering all possible routes. No hard obstacles exist in the environment; therefore, rover movement is allowed in any direction and to any location on the map but attempted traversal of cells that have an expected speed of 0 m/Sol or cells with SD  $\geq$  90 is strongly discouraged due to the potentially infinite ETT due to getting stuck. This is accomplished by setting the travel time to a high, but noninfinite value, such as `intmax` in MATLAB.<sup>37</sup>

The **RRT\*-ETT** Algorithm 1 is an extension of the RRT\* algorithm with altered cost calculation. The `Nearest` function in the standard RRT\* algorithm looks at the nearest node by Euclidean distance, while our function `NearestETT` (Algorithm 2) considers ETT. This step involves solving an integral along the surface, where each step along the surface  $ds$  is multiplied by the expected pace along that step to obtain the ETT. Our environment is uncertain; therefore, we do not assume hard obstacles. Accordingly, we replace the `ObstacleFree` function with the `NoExceed` function (Algorithm 4), which verifies that no cell along the path exceeds the maximum allowable SD, slope, and rock abundance (CFA) for the rover. Within the `NoExceed` conditional statement, the `ParentETT` function (Algorithm 3) considers the closest node by travel time, rather than distance.

### Algorithm 1. RRT\*-ETT.

---

```

T ← InitializeTree()
T ← InsertNode(∅, qinit, T)
for i=1 to i=N do
    qrand ← RandomSample(i)
    qnearest ← NearestETT(T, qrand)
    (qnew, Unew) ← Steer(qnearest, qrand)
    if NoExceed(qnew, qnearest) then
        qnear ← Near(T, qnew)
        (qmin, Umin) ← ParentETT(qnear, qnearest, ...
            qnew, Unew)
        T ← InsertNode(qmin, qnew, T)
        T ← Rewire(T, qnear, qmin, qnew)
    end if
end for

```

---

### Algorithm 2. NearestETT(T, q<sub>rand</sub>).

---

```

for i=1 to i=N do
    TempTT(i) = TravelTime(T(i), qrand)
end for
Return Min(TempTT)

```

---

**Algorithm 3.** ParentETT( $q_{near}$ ,  $q_{nearest}$ ,  $q_{new}$ ,  $U_{new}$ ).

---

```

 $U_{min} = U_{new}$ 
for  $i=1$  to  $i=N$  do
  if TravelTime( $q_{new}, q_{near}(i)$ )  $< U_{min}$  then
     $q_{min} \leftarrow q_{near}(i)$ 
     $U_{min} \leftarrow$  TravelTime( $q_{new}, q_{near}(i)$ )
  else
    Continue
  end if
end for

```

---

**Algorithm 4.** NoExceed( $q_{new}, q_{near}$ ).

---

```

path  $\leftarrow$  DiscretizePath( $q_{new}, q_{near}$ )
for  $i=1$  to  $i=\text{Length}(\text{path})$  do
  if path( $i$ )  $<$  MaxSD & MaxSlope & MaxCFA then
    Continue
  else
    Return Exceed = 1
    Break
  end if
end for
Return Exceed = 0

```

---

**Standard deviation as a measure of information gain**

Information gain algorithms typically utilize Shannon entropy or KL divergence to measure uncertainty. While both Shannon entropy and KL divergence can measure information gain, they are not always appropriate. Shannon entropy is unable to distinguish between different weighted distributions because it only considers raw information gain. KL divergence measures the deviation of a sampled distribution from an ideal distribution, but this introduces bias toward the ideal distribution. This feature of KL divergence is often desirable. However, if the objective is to find regions, which offer the most potential travel time loss or gain, then SD provides a superior measure of information gain. SD-based information gain ensures that regions with broad bimodal probability distributions are targeted over regions with narrow probability distributions. When the probability distribution is heavily weighted at either extreme, the true rover travel time will either be very low or very high. Therefore, regions with broad bimodal distributions offer the greatest potential delta between the expected and true travel times.

Using a separate agent (e.g. helicopter) to observe the grid in advance of the rover's arrival allows several changes to the algorithm. First, the need to obtain information as early as possible during the path plan is relaxed. Next, replacing the Shannon entropy with SD allows the reward component to be simplified to the weighted sum of the reduction in SD over the path. This results in equation (11), used in the following section

$$U_{i,p,t} = \sum_{j=0}^{\text{End}} \left[ W_{j,1} (\text{SD}_{\text{prior}} - \text{SD}_{\text{posterior}}) \right] - W_{i,2} \text{cost}_{i,t} \quad (11)$$

for agent  $i$ , path  $p$ , cell  $j$ , and time  $t$ .

**RRT\*-information theoretic**

The Mars helicopter uses the information-theoretic extension of the RRT\* algorithm presented in this subsection for path planning. The utility function-based algorithm considers a single-agent gaining information as quickly as possible to improve its travel time. This requires obtaining the maximum amount of information early in the trip while balancing the cost of obtaining that information in terms of added travel time. In the case of the rover-helicopter team, the problem has been extended to a multiagent case, where the single-agent seeking to gain information is the helicopter. The information gathered by the helicopter is beneficial to reducing travel time for the rover. In a real-world scenario on Mars, the helicopter can be assumed to travel much faster than the rover and its flight times are extremely short (several minutes) relative to the time scales involved in moving the rover (<200 m/24.66 h). Therefore, the process can be approximated as a two-stage process, or a series of two-stage processes, in which the helicopter first obtains information and the rover then proceeds along its path.

In the case of the rover-helicopter team, the helicopter is assumed to be free to obtain the maximum amount of information available in the nearby area surrounding the start position. Because of the lack of a defined goal location for the information-theoretic helicopter in some cases and the intractability of solving for all possible paths on large-scale maps, an RRT framework called **RRT\*-IT** is used. The **RRT\*-IT** Algorithm 5 considers the utility function 12 in **NearestIT** (Algorithm 6) and **ParentIT** (Algorithm 7) when constructing the tree

$$U_{i,p,t} = W_{i,2} \text{cost}_{i,t} - \sum_{j=0}^{\text{End}} \left[ W_{j,1} (\text{SD}_{\text{prior}} - \text{SD}_{\text{posterior}}) \right] \quad (12)$$

The utility function in equation (12) is modified from equation (11) such that it can be minimized because maximizing a cost function is not possible with RRT-based algorithms. Therefore, finding the paths which minimize equation (12) between any two points will generate optimal paths.

**Helicopter communications and sensor model**

This research is intended to highlight a method for using a helicopter to improve the travel time of a rover and to provide an upper bound on the improvements in travel

time. Communication is not directly modeled, however, without communication between the two vehicles, the rover will revert to using its original RRT\*-ETT routing solution with only the satellite imagery as an input. The rover's travel time in this case will match the safe rover distribution shown in Figure 12.

**Algorithm 5.** RRT\*-IT.

---

```

T ← InitializeTree()
T ← InsertNode(∅,  $q_{init}$ , T)
for i=1 to i=N do
     $q_{rand} \leftarrow \text{RandomSample}(i)$ 
     $q_{nearest} \leftarrow \text{NearestIT}(T, q_{rand})$ 
     $(q_{new}, U_{new}) \leftarrow \text{Steer}(q_{nearest}, q_{rand})$ 
     $q_{near} \leftarrow \text{Near}(T, q_{new})$ 
     $(q_{min}, U_{min}) \leftarrow \text{ParentIT}(q_{near}, q_{nearest}, q_{new}, U_{new})$ 
    T ← InsertNode( $q_{min}$ ,  $q_{new}$ , T)
    T ← Rewire(T,  $q_{near}$ ,  $q_{min}$ ,  $q_{new}$ )
end for

```

---

**Algorithm 6.** NearestIT( $T, q_{rand}$ ).

---

```

for i=1 to i=N do
    TempU(i) = Utility( $T(i), q_{rand}$ )
end for
Return Min(TempU)

```

---

**Algorithm 7.** ParentIT( $q_{near}, q_{nearest}, q_{new}, U_{new}$ ).

---

```

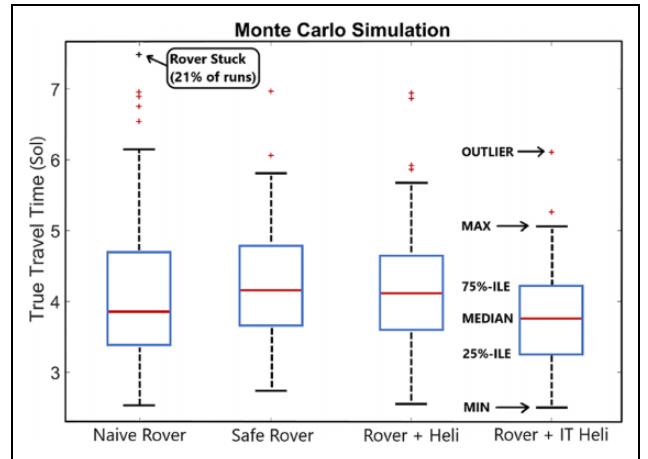
 $U_{min} = U_{new}$ 
for i=1 to i=N do
    if Utility( $q_{near}(i), q_{new}$ ) <  $U_{min}$  then
         $q_{min} \leftarrow q_{near}(i)$ 
         $U_{min} \leftarrow \text{Utility}(q_{near}(i), q_{new})$ 
    else
        Continue
    end if
end for

```

---

Additionally, the communication assumption is reasonably safe given that the rover is capable of moving no more than 200 m in a 24.66 h time period and the helicopter has flight times of several minutes before needing to recharge using solar energy. Therefore, the distances between the two vehicles are assumed to be no more than a few hundred meters at any given time allowing for reliable communication over the relatively short distances separating them.

Our helicopter model has sensors with a viewing cone originating from an altitude of approximately 50 m above the surface. Information gain for the helicopter is modeled by generating a ground truth data set, which is revealed when the helicopter observes a location. A simple sensor model can be implemented in which the helicopter has a



**Figure 12.** Results of Monte Carlo simulation on a user-defined region (Figure 13) displayed as a box plot. True travel time is in Martian Sol (1 Sol  $\approx$  24.66 h). The IT helicopter provides safe approximately 10% median travel time savings or 2.3 h per Sol.

measurement error and the SD is reduced to a lesser but nonzero value.

For an initial distribution from a satellite image observation of location  $i$  described by the mean  $\mu$  and variance  $\sigma^2$

$$P_{\text{sat},i} = [\mu_{\text{sat},i}, \sigma_{\text{sat},i}^2] \quad (13)$$

and a subsequent helicopter observation of location  $i$

$$P_{\text{heli},i} = [\mu_{\text{heli},i}, \sigma_{\text{heli},i}^2] \quad (14)$$

an estimate of the true distribution is given by

$$P_{\text{est},i} = \left[ (1 - \beta)\mu_{\text{sat},i} + \beta\mu_{\text{heli},i}, (1 - \beta)\sigma_{\text{sat},i}^2 + \beta\sigma_{\text{heli},i}^2 \right] \quad (15)$$

where

$$\beta = \frac{\sigma_{\text{sat},i}^2}{\sigma_{\text{sat},i}^2 + \sigma_{\text{heli},i}^2} \quad (16)$$

A sensor model for the Mars helicopter was not implemented in our research for several reasons. First, the path-planning decisions of the helicopter are intended to maximize information gain for the rover and should not be affected by sensor error. The locations of highest uncertainty are still the best locations to observe, even if the helicopter's observations are imperfect. Unlike POMDP models which typically try to account for localization measurement errors when planning a path, our helicopter model uses a more efficient RRT\*-based approach which finds paths that maximize the potential for information gain to benefit the rover. A helicopter sensor model would serve merely as an intermediate data processing step between the helicopter's path-planning stage and the rover's path-planning stage. This intermediate step would not change the routing decisions of the information-theoretic helicopter.

Second, we assume that the helicopter's observations do not increase uncertainty. The rover + IT heli model (Figure 12) represents a best-case solution in which the helicopter is able to remove uncertainty in the observed regions. The safe rover model (Figure 12) represents the worst-case scenario in which the rover must navigate based on satellite imagery containing quantifiable error. As error of the helicopter sensors increases, the resulting rover travel time distribution is expected to lie somewhere between these two bounds.

Third, errors in the helicopter's observations do not reflect on the performance of the safety-oriented rover path-planning algorithm we have developed, because the safe rover (Figure 12) path-planning algorithm already accounts for error in the observations when planning its path. If the helicopter does not provide perfect observations, then the rover's routing algorithm accounts for this by planning the best-case safety-oriented path. Travel time improvements for the rover can be expected to be reduced as known helicopter sensor error increases, just as they do given a known stereoscopic satellite image error.

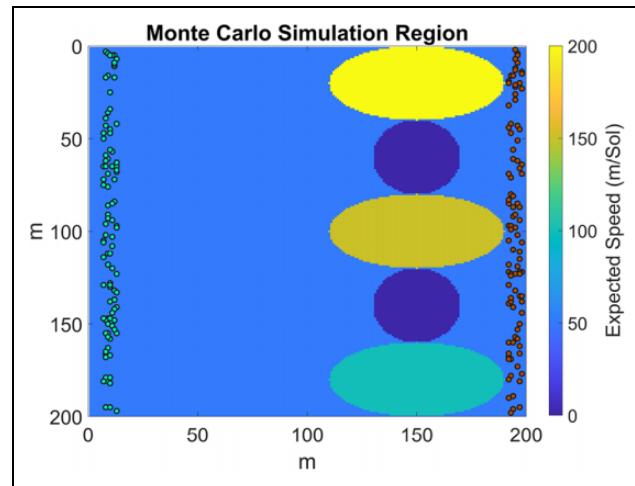
Fourth, considering the effects of unknown helicopter sensor error due to a sensor model, which gives inaccurate information without any quantification of error or SD, is beyond the scope of this research. For example, we do not consider scenarios in which the helicopter incorrectly presents 200 m/Sol with 100% certainty, when the true rover speed is actually 0 m/Sol with 100% certainty.

## Results

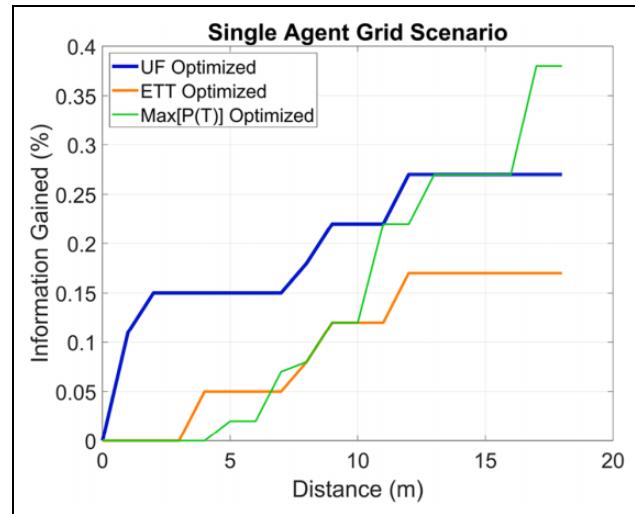
### Scenario environment

Using a utility function based on the Shannon entropy information gain metric in equation (10), an example non-RRT-based scenario (depth-first search) is run on the user defined grid in Figure 5. The results of this example scenario are shown in Figures 14 and 15. In a highly correlated uncertain environment, a single agent benefits most by gaining information as early as possible during its trip. This allows the agent to exploit the information gain to reduce its own travel time. In the case of Figure 15, the utility function optimized path gains more information without sacrificing travel time cost. Subsequent agents traversing this grid can exploit the greater information gained by the first agent using the utility function optimized path. Information gained at the end of the trip is less likely to provide significant travel time savings for a single agent because that agent is too far along the path to be able to use the information gained to reduce its own travel time. Results show that by prioritizing information gain early in the trip, a single agent is free to exploit that information gain during the rest of its trip.

The information gained by the agent can provide significant travel time savings over the initially planned path if

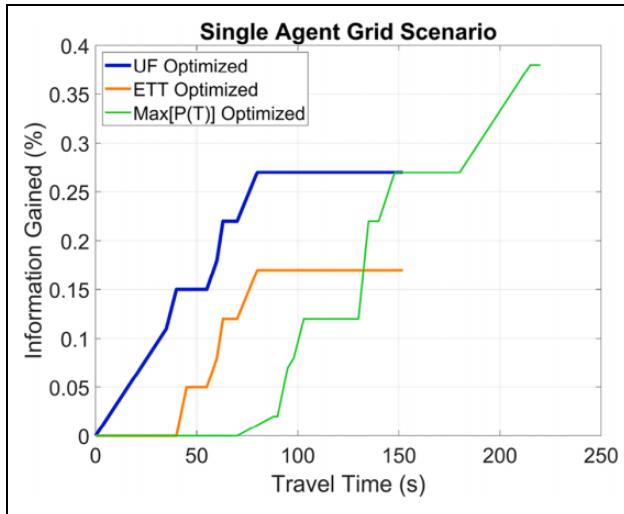


**Figure 13.** User-defined region with the 100 start (green, left) and goal (red, right) locations used in each Monte Carlo simulation. Parameters of the three elliptical regions also vary for each MC run.



**Figure 14.** Single agent: Comparing information gain (y-axis) over distance (x-axis) along three paths using the developed **utility function**, only minimizing **expected travel time**, and assuming the **max[P(T)] travel time** is the true travel time for each cell.

the initial path plan traverses highly uncertain cells. Another benefit of this methodology is that it enables a single agent to perform more nondriving related (e.g. scientific exploration) tasks without a substantial increase in true travel time over the ETT. For a Mars rover, this could mean being able to visit and observe more types of terrain and gain significantly more scientific information while still arriving at the goal location on time. In the following sections, RRT\*-based algorithms are evaluated. The helicopter is the first agent to traverse the map using the **RRT\*-IT** algorithm, while the rover benefits from this information by planning a path, which is more optimal.



**Figure 15.** Single agent: Comparing information gain (y-axis) over travel time (x-axis) along three paths using the developed **utility function**, only minimizing **expected travel time**, and assuming the **max[P(T)] travel time** is the true travel time for each cell. The optimal path in Figure 5 uses the developed utility function, which maximizes information gain while minimizing travel time.

### Simulation environment

The Mars environment is complex and has the potential to introduce terrain-specific geometric bias based on the selected simulation region. This could result in strong bias toward certain routes due to the steepness of the terrain or distribution of rocks, reducing the apparent effectiveness of our approach. Prior to testing the model on the Mars data set, a separate user-defined region was generated to avoid the problem of bias.

A Monte Carlo simulation of 100 runs for each of the four scenarios was performed on a user-defined region with two circular obstacles and three elliptical passages around the obstacles (Figure 13). Each of the three passages was randomly assigned an expected speed, SD, and ground truth for each run. The ground truth is revealed only in locations, which have been observed directly by the helicopter's view cone. The start and goal locations are drawn from a uniform random distribution covering the extent of the  $y$ -axis but fixed over a small range of  $x$ -axis values. This ensures that the rover must always pass by the obstacles by traversing one of the passages and removes bias induced by the geometry of the environment. The helicopter is assumed to have an altitude-dependent observation radius. The viewing radius is modeled by projecting a viewing cone from the helicopter in the  $-z$  direction. The intersection of the viewing cone with the surface defines the outer boundary of the observable area.

Figure 12 shows results for the four scenarios. The naive rover assumes a simple RRT\* algorithm, which does not consider risk. This algorithm performs well when conditions allow because it attempts to find the shortest path to

the goal, however, it does not consider the risk of traversing highly uncertain regions and gets stuck 21% of the time. Note that the box plot for the Naive Rover scenario is calculated after removing the 21% of points with infinite travel time. The safe rover scenario uses the **RRT\*-ETT** algorithm to ensure that the rover safely reaches the destination by avoiding regions of high uncertainty. Because the safe rover avoids regions of very high SD, it typically avoids surprises. However, occasionally, the rover will encounter a significant delay.

The safe rover is generally more cautious than the rover + IT heli team because of lower travel time certainty due to the lack of helicopter scouting. The worst-case performance of this scenario is still vastly superior to the naive rover, which can easily become stuck in the worst case. Under the specific condition that the shortest route is also the fastest route, the naive rover algorithm is capable of achieving good results. However, the naive RRT\* algorithm does not perform well if the shortest route traverses areas of high uncertainty. Caution should be used when making comparisons with the naive rover box plot in Figure 12. There were 21 runs in which the rover became stuck, meaning the travel time is infinite. Therefore, the naive rover box plot is only considering the results in which the rover actually reached the goal. With the other three algorithms in Figure 12, the rover reached the goal in all cases. The best-case results for the naive algorithm are good because the algorithm is optimal in the specific case in which the shortest path is also the fastest path and these cases do arise in the Monte Carlo simulation. For cases in which this does not hold, the naive algorithm risks complete mission failure.

Results show that the presented rover + IT heli algorithm improves the worst-case results significantly compared with the naive shortest path approach. The addition of the IT heli also offers approximately 10% median travel time improvement compared with the safe rover alone in the Monte Carlo simulation region. In the case of long-term missions on Mars, which span months or years, this saving is significant. The rover + IT heli algorithm provides the most robust results, with the lowest travel time SD and the lowest median travel times.

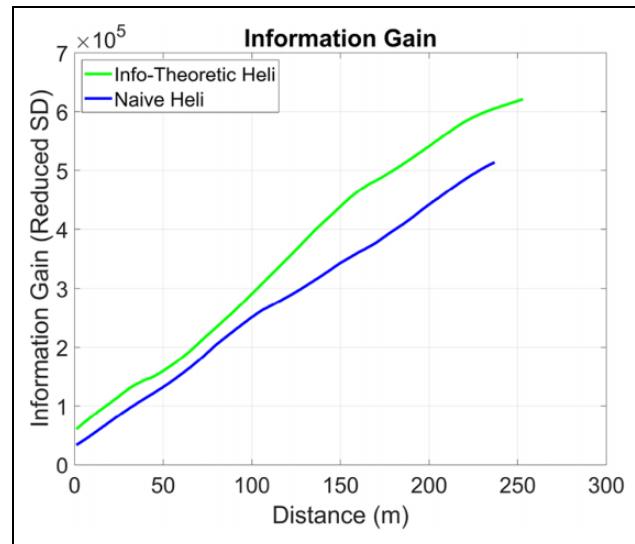
When the rover has a helicopter scouting its planned path, the median result is very similar to when the rover travels alone. The rover's path is already safety-oriented and this is accomplished by avoiding regions of high uncertainty. The naive helicopter which scouts ahead on the rover's planned path fails to improve travel time substantially because it also avoids high uncertainty regions. The result of this technique is a slight improvement in median rover travel time. Small improvements are seen in situations, where the helicopter detects that the speed through the selected passage is less than expected and the rover chooses to minimize its time spent inside the elliptical passage.

When the helicopter can scout regions of high SD, it either uncovers a superior path or discovers that the current path is acceptable. This method is more robust compared to the naive helicopter, because in the case that the rover's planned path passes through the region with the highest SD, the information-theoretic helicopter will scout along the planned path. The rover + IT heli model, therefore, tends to produce the most optimal results under more uncertain conditions.

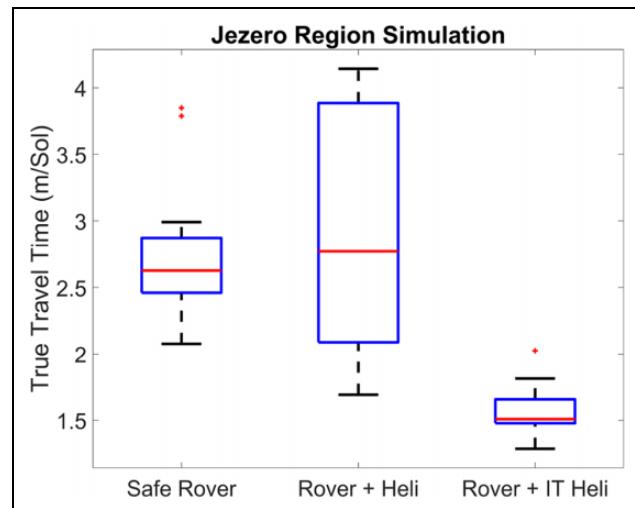
### Jezero region case study

After confirmation of the model using the Monte Carlo simulation, a case study was performed in the Jezero region on Mars. Because of the complexity of the 3D Mars environment and the number of unfeasible start and goal locations, we can select a start and goal location, which offer two main traversable routes to go around a steep cliff. The performance of the rover + IT heli model was compared to the rover alone and to the rover + naive heli model, where the helicopter scouts ahead along the rover's planned route. In this region, the rover + IT heli model finds a superior path for the rover compared to both the rover alone and the rover + naive heli model. Figure 7 shows the results of the case study, where the rover + IT heli model reduced the rover's true travel time from 2.62 Sol to 1.68 Sol.

In this case study, the rover must choose either to go left or right around a region of low traversability caused by an extremely steep slope. The path on one side of the obstacle presents a longer distance for the rover due to the shape of the terrain surface while the path on the other side presents a shorter distance. Because the uncertainty on the longer path is high, the rover will not choose to take this path. Attempting to observe this environment with the rover alone carries a high risk since the region may turn out to increase travel time substantially. The cost of visiting this region and becoming stranded, or turning around and going back, is too great to risk sending the rover to this destination. Therefore, the rover alone will always choose the safer option. The results for this case study are consistent with the Monte Carlo simulation environment with the exception of the Mars helicopter naively observing the path the rover has chosen. In this case, it does not search regions of high uncertainty, since the rover will avoid these regions for those with more reliable travel times. Therefore, the Mars helicopter will only be evaluating regions with lower uncertainty. Using the Mars helicopter, this way tends to produce small median travel time improvements and can cause increased travel time SD for the rover. The rover + IT heli model consistently produces the best results by searching regions of higher SD and reducing uncertainty, as shown in Figure 16. If the travel time can be improved, the rover will adjust its path through the previously high uncertainty region. Overall, the rover + IT heli model



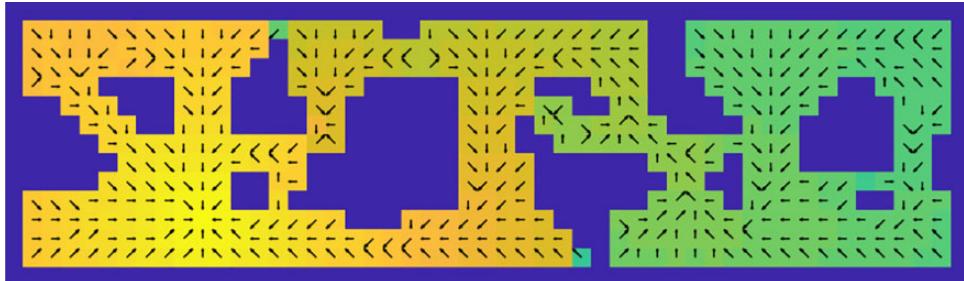
**Figure 16.** Case study on Jezero region showing information gain in terms of the reduction of standard deviation per step along the grid shown in Figure 7 for the **naive helicopter** and the **information-theoretic helicopter**. Information-theoretic helicopter gathers more important information sooner along its path, enabling more effective rerouting of the rover.



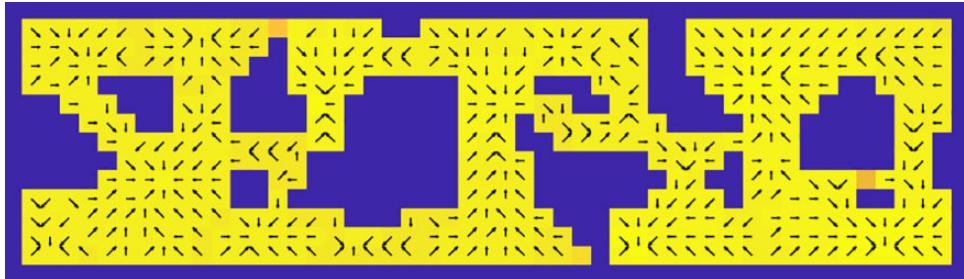
**Figure 17.** Results of 20 run simulation on Jezero region (Figure 7) with constant start and goal location and varying ground truth. True travel time is in Martian Sol (1 Sol  $\approx$  24.66 h). The IT helicopter provides safe median travel time savings of 11.4 h per Sol over the rover alone.

provides greater information gain and travel time savings without increased risk for the rover.

A 20 run simulation was performed on the Jezero region with randomly generated ground truth data and fixed start and goal locations. Due to bias caused by the geometry of the region, when the helicopter naively scouts ahead on the same path, this tends to worsen the rover's performance. This is because the rover only has two feasible routes available and tends to favor one of



**Figure 18.** MDP value iteration solution for a stochastic policy with entropy as a reward and a defined goal state. Yellow is high value, green is low value, where value is a weighted reward.



**Figure 19.** MDP value iteration solution for a stochastic policy with entropy as a reward and an undefined goal state. Yellow is high value and green is low value, where value is a weighted reward.

them due to the geometry of the environment. If the naive helicopter discovers that the preferred route is less optimal than the satellite data indicated, the rover tries the other route, which turns out in some cases to be slower than predicted and due to its greater path length, can add significant travel time cost. In the IT heli case, the rover avoids switching from its preferred route unless the IT heli discovers a significant improvement in travel time on the alternative route. The results for this simulation are shown in Figure 17.

### Scalability

Testing with traditional MDP and POMDP approaches proved computationally intractable for the scale and resolution of this problem. For example, an MCTS algorithm was tested using POMDPs.jl on the following grid sizes and resulted in exponentially increasing computational times, as shown in the following table.<sup>15,38</sup>

Grid size	Average compute time (s)
10 × 10	0.4
15 × 15	18.63
20 × 20	106.14

Both the successive approximations of the reachable space under optimal policies (SARSOP) and determinized sparse partially observable tree (DESPOT) algorithms were

also tested using the approximate POMDP planning (APPL) C++ toolkit.<sup>39,40,41</sup> Both algorithms scaled poorly even on small toy problems. The SARSOP algorithm took 389 s to solve the small-scale underwater navigation problem described in prior research.<sup>42</sup> While our results are superior to the performance in the prior research, this difference is likely due to the use of an I7-8700K processor with a 4.3-GHz clock speed, compared to their use of a 2.66-GHz processor. The DESPOT algorithm solved the same underwater navigation problem in 89 s, which is a considerable improvement but still indicative of poor scalability. Value iteration MDP solvers were also tested on toy maps with entropy as a reward for both defined goal states (Figure 18) and undefined goal states (Figure 19), but the solvers tend to become trapped in local maxima and are also computationally intractable for large high-resolution maps.

The user-defined Monte Carlo simulation region in Figure 13 contains 40,000 1-m<sup>2</sup> cells and the Jezero case study region in Figure 7 contains 673,854 1 m<sup>2</sup> cells, considerably greater than the simple cases attempted for the MDP and POMDP approaches. The RRT\*-based approaches presented in this article are sampling-based and allow for solutions on large-scale 3D environments without excessive memory usage or computational requirements, allowing deployment on a large transistor, radiation-hardened ARM-based CPUs. Computational times using a single core of an Intel I7-8700K processor for the most

complex case of the two-stage algorithm (rover + IT heli) on both the Monte Carlo simulation region and the Jezero case study region are shown in the following table.

Simulation	Average compute time (s)
MC Sim region	1262
Jezero region	1648

The computational time is dependent on the number of nodes and the maximum length of each edge in the tree. Compared with traditional RRT\* algorithms, the presented algorithms take more time due to the fact that ETT is calculated between each node and each edge of the tree traverses a nonuniform surface with varying expected speeds. This requires the calculation of a path integral along each edge. Significant improvements are expected by implementing the code in a programming language such as FORTRAN, which offers much faster looping compared with MATLAB.<sup>43,37</sup>

## Conclusion

This research offers a set of novel techniques for single and multiagent path planning in uncertain environments. The addition of an information-theoretic helicopter guided by the **RRT\*-IT** algorithm allows safe information gain for a ground-based rover without excessive computational cost, and the use of **RRT\*-ETT** algorithm ensures that the rover takes the fastest route without incurring substantial risk. The robustness of the **RRT\*-ETT** algorithm is demonstrated in the Monte Carlo simulation, which shows that even without the helicopter, the rover achieves good median travel times. When the helicopter updates information, the rover takes advantage of the new information while considering the error and updates its path. This methodology provides better routing and extends travel distance allowing more time for the rover to perform nondriving research activities.

By formulating utility functions, which balance the trade-off between exploration and exploitation, this research develops an algorithm for time-dependent path planning with single or multiple autonomous agents. In the model framework, each grid cell (e.g. image pixel) contains the unique probabilistic distribution of travel time, allowing the formulation of path plans under a partial information environment. Invaluable knowledge and insights are derived regarding correlation between cells of the grid environment and integrating different sources of information gain.

## Future research

In real-world applications, the assumption that the helicopter perfectly observes the environment, bringing the SD to zero for all observed cells, is not necessarily valid. Future

research should consider the effects of partial or inaccurate information gain by the helicopter by modifying the rover's path planning algorithm to account for incorrect information.

Additionally, the rover + heli models do not assume correlation between regions in the environment. Significant correlation may exist between regions of similar terrain features. In a highly correlated environment, observing one region can provide information about other regions, which have not yet been observed. Future research will consider this possibility by creating a terrain correlation model, which demonstrates that locations, which share the same terrain type probability distribution are highly correlated. Under this correlation model, information gain obtained by observing one location can reduce uncertainty in another unobserved location. This technique could also be used to adjust travel time predictions by inferring the accuracy of satellite observations. Previous work involving correlation has mostly focused on upstream or downstream effects in a road network, rather than travel time correlation based on satellite imagery.<sup>44,45</sup>

This model framework can be further applied to contribute to multiagent systems with core principles for information sharing. In a disaster situation, when part of a road network is disconnected and traversability is uncertain, the proposed concept can efficiently guide semiautonomous and autonomous rescue vehicles. Future autonomous electric vehicles can incorporate this model with the objective of maximizing energy efficiency, considering the trade-off between energy efficiency and congestion and making better decisions when the outcomes are correlated across the map or the actions of other agents.

## Acknowledgements

The authors express thanks to all editors and reviewers of this manuscript.

## Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partially carried out at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with the National Aeronautics and Space Administration [80NM0018D0004], and partially at the North Carolina Agricultural and Technical State University under a contract with the Jet Propulsion Laboratory, California Institute of Technology [JPL RSA 1625294] and NSF RI 1910397. The additional partial support was provided by the North Carolina Department of Transportation TAR 2019-09, Virginia Department of Transportation VTRC 116038, and USDOT University Transportation Centers [Contract 69A3551747125].

**ORCID iD**Hyoshin Park  <https://orcid.org/0000-0002-1490-5404>**References**

- Shannon CE and Weaver W. *The mathematical theory of communication*. Urbana: The University of Illinois Press, 1949.
- Kullback S and Leibler RA. On information and sufficiency. *Ann Math Stat* 1951; 22(1): 79–86.
- Das Gupta M, Srinivasa S, Madhukara J, et al. KL divergence based agglomerative clustering for automated vitiligo grading. In: *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015, pp. 2700–2709. Los Alamitos: IEEE. DOI: 10.1109/CVPR.2015.7298886.
- Ono M, Rothrock B, Cunha de Almeida E, et al. Data-driven surface traversability analysis for Mars 2020 landing site selection. In: *2016 IEEE aerospace conference*, Big Sky, MT, USA, 5–12 March 2016. pp. 1–12. Los Alamitos: IEEE. DOI: 10.1109/AERO.2016.7500597.
- Rothrock B, Kennedy R, Cunningham C, et al. SPOC: deep learning-based terrain classification for Mars rover missions. In: *American Institute of Aeronautics and Astronautics (AIAA) SPACE 2016*. 2016. DOI:10.2514/6.2016-5539.
- McEwen AS, Eliason EM, Bergstrom JW, et al. Mars reconnaissance orbiter's high resolution imaging science experiment (HiRISE). *J Geophys Res Planets* 2007; 112(E5). DOI:10.1029/2005JE002605.
- Golombek M, Huertas A, Kipp D, et al. Detection and characterization of rocks and rock size-frequency distributions at the final four mars science laboratory landing sites. *MARS: The International Journal of Mars Science and Exploration* 2012; 7: 1–22.
- Ono M, Heverly M, Rothrock B, et al. Mars 2020 site-specific mission performance analysis: Part 2. Surface traversability. In: *2018 AIAA SPACE and astronautics forum and exposition*, Orlando, FL, 17–19 September 2018. DOI:10.2514/6.2018-5419.
- Ono M, Rothrock B, Otsu K, et al. Maars: Machine learning-based analytics for rover systems. In: *IEEE aerospace conference 2020*, Big Sky, MT, USA, 7–14 March 2020.
- Moldovan TM and Abbeel P. Safe exploration in Markov decision processes. In: *Proceedings of the 29th international conference on machine learning* (eds J Langford and J Pineau), Edinburgh, Scotland, UK, 27 June–3 July 2012, pp. 1451–1458. Madison: Omnipress.
- Delmerico J, Mueggler E, Nitsch J, et al. Active autonomous aerial exploration for ground robot path planning. *IEEE Robot Autom Lett* 2017; 2(2): 664–671.
- Babel L and Zimmermann T. Planning safe navigation routes through mined waters. *Eur J Oper Res* 2015; 241(1): 99–108.
- Yakovlev K, Andreychuk A, Belinskaya J, et al. Combining safe interval path planning and constrained path following control: Preliminary results. In: *Proceeding of the 4th international conference on interactive collaborative robotics (ICR 2019)* (eds A Ronzhin, G Rigoll and R Meshcheryakov), 20–25 August 2019, pp. 310–319. Cham: Springer International Publishing.
- Delamer JA, Watanabe Y, and Ponzoni Carvalho Chanel C. MOMDP solving algorithms comparison for safe path planning problems in urban environments. In: *10th Workshop on planning, perception and navigation for intelligent vehicles*, Madrid, Spain, 2018, pp. 1–6. Toulouse, France: Open Archive Toulouse Archive Ouvert (OATAO).
- Coulom R. Efficient selectivity and backup operators in Monte-Carlo tree search. In: *Computers and games CG 2006* (eds HJ van den Herik, P Ciancarini and HHLM Donkers), Turin, Italy, 29–31 May 2006. Berlin, Germany: Springer-Verlag.
- Junges S, Jansen N, Wimmer R, et al. Finite-state controllers of POMDPs via parameter synthesis. In: Globerson A (ed) *Uncertainty in artificial intelligence: 34th conference*. Corvallis: AUAI Press, 2018, pp. 519–529.
- Balaban E, Roychoudhury I, Spirkovska L, et al. Dynamic routing of aircraft in the presence of adverse weather using a POMDP framework. In: *17th AIAA aviation technology, integration, and operations conference*, 5–9 June 2017, Denver, Colorado.
- Qiao Z, Mülling K, Dolan J, et al. POMDP and hierarchical options MDP with continuous actions for autonomous driving at intersections. In: *2018 21st IEEE international conference on intelligent transportation systems (ITSC)*, Maui, HI, USA, 4–7 November 2018, pp. 2377–2382. Los Alamitos: IEEE. DOI: 10.1109/ITSC.2018.8569400.
- Garg NP, Hsu D, and Lee WS. DESPOT-alpha: Online POMDP planning with large state and observation spaces. In: Bicchi A, Kress-Gazit H, and Hutchinson S (eds) *Robotics: science and systems XV*, Freiburg im Breisgau, Germany, University of Freiburg, 2019.
- BAE Systems. BAE systems radiation-hardened electronics, space systems - processor products. <https://www.baesystems.com/en-us/product/radiation-hardened-processors-products> (2020, accessed 09 October 2020).
- Gao X, Wu H, Zhai L, et al. A rapidly exploring random tree optimization algorithm for space robotic manipulators guided by obstacle avoidance independent potential field. *Int J Adv Robot Sys* 2018; 15(3). DOI:1729881418782240.
- Connell D and La HM. Extended rapidly exploring random tree-based dynamic path planning and replanning for mobile robots. *Int J Adv Robot Sys* 2018; 15(3). DOI:1729881418773874.
- Sengupta S. A parallel randomized path planner for robot navigation. *Int J Adv Robot Sys* 2006; 3(3): 37.
- Silva MC, Garcia ACB, and Conci A. A multi-agent system for dynamic path planning. In: *2010 second brazilian workshop on social simulation*, Sao Paulo, Brazil, 24–25 October 2010, pp. 47–51. Los Alamitos: IEEE. DOI:10.1109/BWSS.2010.20.
- Martin S, Ouelhadj D, Beullens P, et al. A multi-agent based cooperative approach to scheduling and routing. *Eur J Oper Res* 2016; 254(1): 169–178.

26. Yoon S and Kim J. Efficient multi-agent task allocation for collaborative route planning with multiple unmanned vehicles. *IFAC-PapersOnLine* 2017; 50(1): 3580–3585.
27. Ziebart BD, Bagnell JA, and Dey AK. Maximum causal entropy correlated equilibria for Markov games. In: *Proceedings of the 3rd AAAI conference on interactive decision theory and game theory*, AAAIWS'10-03. AAAI Press, 2010, pp. 74–80.
28. Chen F. Aircraft taxiing route planning based on multi-agent system. In: *2016 IEEE advanced information management, communicates, electronic and automation control conference (IMCEC)* (ed B Xu), Xian, China, 3–5 October 2016. Los Alamitos: IEEE. pp. 1421–1425.
29. Lam CP, Yang AY, Driggs-Campbell K, et al. Improving human-in-the-loop decision making in multi-mode driver assistance systems using hidden mode stochastic hybrid systems. In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Hamburg, Germany, September 28–October 02 2015, pp. 5776–5783.
30. Waddell D, Pugh N, Shirzad K, et al. Simulation-based optimization of emergency response considering rationality of travelers. In: *Proceeding of the 98th annual meeting of transportation research board (TRB 2019) Washington, DC*, 2019, Paper No. 19-05975.
31. Waddell D, Pugh N, and Park H. Visualization-based dynamic dispatching of first responders. In: *Proceeding of the 98th annual meeting of transportation research board (TRB 2019)* Washington, DC, 2019, Paper No. 19-05569.
32. Park H and Haghani A. Optimal number and location of Bluetooth sensors considering stochastic travel time prediction. *Transp Res Part C Emerg Technol* 2015; 55: 203–216.
33. Park HJ, Haghani A, Gao S, et al. Anticipatory dynamic traffic sensor location problems with connected vehicle technologies. *Transp Sci* 2018; 52(6): 1299–1326.
34. GDAL Development Team. *GDAL - Geospatial data abstraction library, version 3.0.1*. Open Source Geospatial Foundation, 2019, <http://www.gdal.org> (2019, accessed 13 August 2019).
35. Lavalle SM. Rapidly-exploring random trees: a new tool for path planning. Technical Report No. 98-11, Computer Science Department, Iowa State University 1998; 1–4.
36. Abbadi A, Matousek R, Jancik S, et al. Rapidly-exploring random trees: 3D planning. In: *18th international conference of soft computing, MENDEL 2012*. Brno, Czech republic, 27–29 June 2012. Brno: VUT Press. DOI: 10.13140/2.1.3632.3848.
37. MATLAB. *MATLAB and signal processing toolbox release 2019b*. Natick, MA: The MathWorks Inc, 2019.
38. Egorov M, Sunberg ZN, Balaban E, et al. POMDPs.jl: a framework for sequential decision making under uncertainty. *J Mach Learn Res* 2017; 18(26): 1–5. <http://jmlr.org/papers/v18/16-300.html> (accessed 17 September 2020).
39. Kurniawati H, Hsu D, and Lee WS. SARSOP: efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: *Robotics: science and systems IV, Eidgenössische technische hochschule Zürich* (eds O Brock, J Trinkle and F Ramos), Zurich, Switzerland, 25–28 June 2008. Cambridge: MIT Press Ltd.
40. Ye N, Somani A, Hsu D, et al. DESPOT: online POMDP planning with regularization. *J of Art Intell Res* 2017; 58(1): 231–266.
41. Smith T, Cassandra T, and Thomason L. *APPL - Approximate POMDP planning C++ toolkit, version 0.96*, 2015. <https://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/index.php?n>Main.FAQ> (accessed 18 September 2020).
42. Ong S, Png S, Hsu D, et al. POMDPs for robotic tasks with mixed observability. In: *Robotics: science and systems V*. Robotics: Science and Systems Foundation, 2009. DOI:10.15607/rss.2009.v.026.
43. Ellis TMR, Philips IR, and Lahey TM. *Fortran 90 programming*. Boston: Addison-Wesley, 1994.
44. Frejinger E and Bierlaire M. Capturing correlation in route choice models using subnetworks. *Transport Res B-Meth* 2007; 41: 363–378.
45. Xing T and Zhou X. Finding the most reliable path with and without link travel time correlation: a Lagrangian substitution based approach. *Transp Res Part B: Methodol* 2011; 45(10): 1660–1679.