SAMPLING-BASED DESCENT TRAJECTORY PLANNING AND AUTONOMOUS LANDING SITE SELECTION FOR ICY MOON LANDER MISSIONS

Akash Arora[1], P. Michael Furlong[2], Uland Wong[2], Terrence Fong[2]

[1]*Australian Centre for Field Robotics, The University of Sydney, Australia, E-mail: aaro4138@uni.sydney.edu.au*
[2]*Intelligent Robotics Group, NASA Ames Research Centre, CA, USA, E-mail:*
{padraig.m.furlong,uland.wong,terry.fong}@nasa.gov

ABSTRACT

Selecting suitable landing sites is fundamental to achieving success in robotic lander missions. However, due to sensing limitations, landing sites that are both safe and scientifically suitably may not be determined reliably from orbit prior to descent, especially where orbital sensing data is noisy or incomplete. In previous work we proposed an algorithm which allows landers to autonomously plan informative descent trajectories, exploiting information gathered during descent to land on high quality sites [1]. In this paper, we improve the scalability of our approach through importance sampling. Our method substantially reduces planning times without sacrificing the quality of the selected landing sites.

1 INTRODUCTION

The icy moons of Europa and Enceladus are among the top priorities for NASAs exploration objectives [2]. These bodies may be the best candidates for finding extra-terrestrial life in the solar system, as interior liquid oceans may be present and accessible from the frozen surface. Due to the remote nature of these missions, however, human intervention is limited and on-board autonomy is necessary to execute complex tactical maneuvers during entry, descent, and landing (EDL) and ensure the robot safely lands at the desired site.

In past lander missions, suitable landing sites were selected a priori by domain experts from orbital sensor measurements of geological features that indicate the safety and scientific utility of sites. Low level navigation, control, and hazard avoidance were conducted on-board to ensure a safe landing near the desired site without human intervention. However, noisy, incomplete, or low resolution orbital data can mask the true safety and scientific value of landing sites. This is especially the case for icy moons, where key features such as crevasses, jagged penitentes, liquids, and ice thickness can either be either below the resolution of orbital sensors or require non-traditional sensors, like ground penetrating radar or thermal imaging, to detect. These non-traditional sensors have limited sensing range and cannot necessarily resolve the required information from orbit.

In previous work [1], we presented a new approach to autonomous EDL which allows spacecraft to land in suitable sites under such sensing constraints. We proposed a Bayesian network (BN) architecture that conducts on-board heterogeneous sensor data analysis and rapidly updates estimates of safety and scientific suitability of landing sites during descent. The estimated values were fed into a sampling based planner based on Monte Carlo Tree Search [3] to plan informative descent trajectories such that the lander can acquire high quality observations of promising landing sites during descent subject to fuel constraints. The descent trajectories were updated as new data were acquired and replanned until landing. Our algorithm is anytime and requires constant memory, making it particularly well suited for missions with low computational power and hard real-time constraints.

Previous work evaluated the approach in relatively small landing site maps. This work aims to scale up the approach to larger maps. We conduct a complexity analysis of our approach which we use to target areas where approximations would impact performance. We then propose algorithmic modifications based on importance sampling [4] which significantly improves scalability of our approach to larger environments. Lastly, we present an empirical evaluation of the approach in both large simulated maps with realistic icy terrain and thermal data based on physics models. We complement these extensions with increased technical detail, generalization to arbitrary EDL missions, and additional discussion on the practical challenges of deploying our approach in practice.

2 RELATED WORK

In past lander missions, landing sites were chosen by the science team based on orbital data and domain knowledge. Autonomous EDL research has largely focused on hazard detection using computer vision and navigation techniques to accurately land in a desired location. On-board perception is used to match low-altitude terrain to maps created from orbital data which helps determine if a lander is accurately executing a predetermined trajectory from relative motion [5, 6]. Last minute diversions are allowed to avoid hazardous sites if detected, but unlike our approach, the spacecraft cannot tour multiple candidate landing sites and learn more about the environment before committing to a site. We also consider the scientific utility of sites, not just the safety.

Typical EDL sensor packages include lidar, radar, and cameras to characterize terrain geometry. Interactions of these sensors with icy moon surfaces however could lead to increased sensor noise. Non-traditional sensors like bore-sight imagers, thermal cameras, and sounding radar may need to be used which have varying sensing ranges and noise models. Our approach is sensor agnostic and can incorporate an arbitrary number of sensors with arbitrary noise models.

Serrano et al. proposed a Bayesian network (BN) for planetary landing site selection which fuses data from multiple sensor sources, and other criteria such as reachability and scientific utility to determine probabilistic estimates of quality of landing sites [7]. We also use BNs to fuse multi-modal data but unlike Serrano's work, we account for altitude dependent sensor noise and use the probabilistic estimates to actively explore and gather information about the environment.

Desaraju, et al., explores terrestrial rooftop environments with a UAV to select the best landing site [8]. They use a Gaussian process to model the environment and combine it with information gain functions to explore the environment. The work is similar in principal to ours, but is application specific and does not incorporate multi-modal sensing and the decision of when to use which sensor.

Descent trajectory planning is traditionally viewed from an optimal control perspective where the target location is provided a-priori [9]. In contrast, we plan trajectories using Monte Carlo Tree Search (MCTS) which is a sampling based, approximate tree search algorithm [3]. MCTS is anytime, giving it a unique advantage over both gradient descent and other sampling based approaches like Rapidly Exploring Random Trees. Our adaptation of MCTS allows principled reasoning over long horizons, and accounts for partially observability [10] which is particularly advantageous in an EDL situation where there are hard real time constraints and observations from sensors are noisy or incomplete. Furthermore our approach allows spacecrafts to simultaneously plan movements and schedule when to make sensor measurements to gain further information- a capability beyond the scope of existing research in descent trajectory planning. Sampling based methods, however, inherently discretize the planning space but can be fused with continuous space optimization techniques to ensure smoothness and dynamic feasibility.

3 PROBLEM FORMULATION

The spacecraft must generate descent trajectories and sensor schedules which gather information on promising landing sites and terminate on a site that is both safe and scientifically valuable. This section describes the environment model, the properties of our simulated lander, and formally defines the planning problem being solved.

Environment Representation: This work uses a grid world representation of the environment where each grid cell is a potential landing site. Cells are described by feature vectors, F. On icy moons, these features could include ice thickness, terrain jaggedness, slope, and thermal properties. The lander can take noisy measurements of these features through its on-board sensors.

We assume that there are functions that map feature vectors to safety, $V_T : F \rightarrow \{Unsafe, Safe\}$, and the scientific value, $V_S : F \rightarrow [0, \ldots, \infty)$, of the candidate landing sites. There exists prior work which aims to estimate landing site safety from sensor data, using techniques such as plane fitting, edge detection, and neural network classification [6, 11]. We can use these functions in our approach to supply V_T.

V_S indicates which geological features are high priority for the mission's science goals. This function must be elicited from the mission science team. The overall site utility U is a function of the scientific value and the safety of the site.

Lander Properties: At any given time, the lander

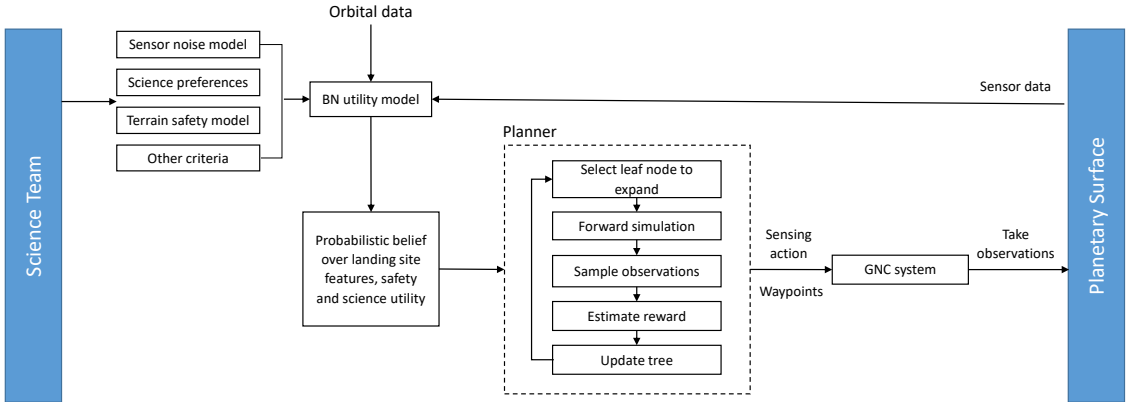

Figure 1: The systems architecture of our approach

can choose a maneuvering action which changes the vehicle's direction of motion and select which of its P-many sensors to use. An example sensor payload for an icy moon lander could include a high resolution camera, ground penetrating radar, thermal sensor, and a reflectance spectrometer. Each sensor observes different subsets of the feature vector F and has its own noise model and field of view which varies with the spacecraft altitude. We discretize the maneuvering space m into K actions. This produces an action space, $A = \{m_1, \ldots, m_K\} \times \{s_0, s_1, \ldots, s_P\}$.

Problem Statement: The lander must plan a sequence of maneuvering and sensing actions $a_{1\ldots L}$ which maximize some reward function R measuring the likelihood of landing at a site with high overall utility. Each maneuvering or sensing action a_i incurs some predefined cost given by the $cost(a_i)$ function and the overall sequence is subject to a budget B. This budget could be the delta-V or time. The optimization problem is stated below:

$$a^*_{1..L} = \arg\max_{a_{1\ldots L} \in A} R(a_{1\ldots L}) \quad \text{s.t.} \sum_{i=1}^{L} \text{cost}(a_i) \leq B \tag{1}$$

Lastly, when the lander budget expires, it must be within some radius L of the selected landing site.

$$|x_f - x_{end}| \leq R \tag{2}$$

where x_{end} is the lander's terminal position and x_f is the desired landing site. It is expected that at this point in a real mission the lander will be very close to the surface, and the powered descent stage will begin and the divert capabilities will be limited to hazard avoidance.

4 OVERVIEW OF APPROACH

Our approach consists of two main components: estimating candidate site utilities, and planning informative descent trajectories. A systems architecture of the approach is illustrated in Fig. 1. This section provides an overview of these components and how they can be applied in real missions.

4.1 Site Utility Estimation

As mentioned in Sec. 3, the environment is discretized into cells where each cell is a potential landing site. To maximize the reward function in Eq. 1 and land on high quality sites, the spacecraft needs a way to infer the safety and science utilities of sites from on-board heteregeneous sensor measurements taken during descent. Since sensor observations are inherently noisy, we probabilistically model the inference of safety and science utility using the Bayes' Net in Fig. 2.

Orbital data is used to initialize Bayesian prior distributions over the geological features $\mathbf{F}$ in each cell. These distributions are updated using data collected during descent from the on-board sensors through observations Z_p, where $p \in \{1, \ldots, P\}$ is the sensor used. There is an independent BN associated with each landing site.

Each sensor measures different subsets of these geological features from which the safety T, science utility S and overall utility U of a landing site can be estimated. In this problem setting we set the Z and $\mathbf{F}$ nodes to be discrete categorical variables as it simplifies inference, S as a continuous, non-negative, variable and T as a variable ranging from 0 to 1 indicating the probability whether a site if safe or not.

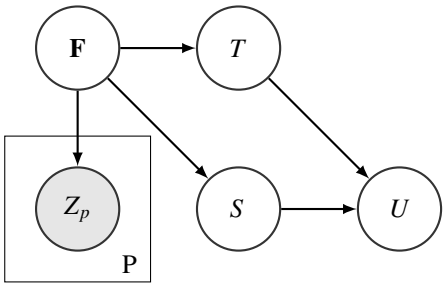

Figure 2: Bayesian network to calculate science utilities based on observations. Observations from the different senors, $Z_1, \ldots, Z_p$ inform the feature vector **F** *which provides information on science value, S, and landing site safety T.*

Given an observation, the overall utility of a site can be recursively estimated by conducting Bayesian updates:

$$\begin{aligned} P(U|Z) &= \sum_{T,S} P(U|T,S,Z).P(T,S|Z) \\ &= \sum_{T,S} P(U|T,S) \sum_{F} P(T|F)P(S|F)P(F|Z) \end{aligned} \tag{3}$$

The conditional probability terms in Eq. 3 quantify the probabilistic relationships between variables. It can be deduced that $P(\mathbf{F}|Z_p)$ defines a feature classification model, while the $P(T|\mathbf{F})$ and $P(S|\mathbf{F})$ terms classify the safety and scientific utility of the site based on the geological features. We now define each of these terms in more detail.

Quantifying Science Utility: Prior to the mission, scientists' preference for desired attributes in landing sites can be formulated as a value function that maps features of a region to some score. In icy moon missions, the features of interest may include presence of bio-markers, proximity to liquids, or desirable thermal properties of ice. We assume that the scientists' utility function is known. We use a weighted linear function of geological features but any arbitrary function can be used.

Quantifying Site Safety: Landing sites need to be classified as either "Safe" or "Unsafe", based on geological features at the site and the design of the lander. We assume this term is provided a priori based on domain knowledge or existing learning and classification techniques [11, 7].

Feature Classification: Discriminative classifiers have been extensively used to classify geologic features in remote sensing data [12]. We use a generative classifier, $P(F|Z_p)$, because our planner requires the ability to predict future observations. We can invert our classifier to produce $P(Z_p|F)$ for planning purposes, using Bayes theorem.

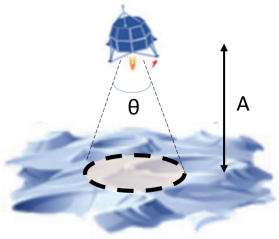

Figure 3: Our sensors have circular fields of view. The viewing cone is characterized by angle θ and spacecraft altitude A.

Sensor Model: We model onboard sensors with a circular field of view, as in Fig. 3. Decreasing altitude increases the resolution of the sensor, and decreases the effects of sensor noise. The sensor noise model for sensor p at altitude A is given by Eq. 4. $RMax$ is the maximum sensing range, $P(Z|\mathbf{F})_{best(p)}$ is the best case sensor noise model and G_p is the distribution of the sensor's intrinsic noise model. For example, a thermal camera G_p may model pink noise, while a laser altimeter G_p may be a uniform distribution. Non-linear noise functions can also be used here.

$$\begin{aligned} P(Z|F)_{p,a} &= \alpha P(Z|F)_{best(p)} + (1-\alpha)G_p \\ \alpha &= \begin{cases} 0, & \text{if } A \geq RMax(p) \\ 1 - \frac{A}{RMax(p)}, & 0 \leq A < RMax(p) \\ 1, & A < 0 \end{cases} \end{aligned} \tag{4}$$

We assume $RMax$, $P(Z|\mathbf{F})_{best(p)}$, G_p, and the type of features seen depend on the sensor type and are known a priori. Sensor measurements taken throughout the mission are fed into the BNs to recursively update features estimates in each observed landing site. Updated feature descriptors are then used to predict safety and science utilities along with the uncertainty using Eq. 3. We use particle filters and message passing to propagate the belief updates through the network.

4.2 Planning Descent Trajectories

According to decision making theory, the optimal sensing action sequence is one which in *expectation* terminates at the highest utility (U) landing site. We define the reward function to optimize

$R(\cdot)$ as:

$$R(a_{1\ldots L}) = \sum_{Z_{1\ldots L}} \mathbb{E}(U(x_f|Z_{1\ldots L}))P(Z_{1\ldots L}|a_{1\ldots L}) \quad (5)$$

$Z_{1\ldots L}$ are the observations made by taking actions $a_{1\ldots L}$, $P(Z_{1\ldots L}|a_{1\ldots L})$ is the probability distribution of observations that can be made given a sensing sequence, and $U(x_f|Z_{1\ldots L})$ is a mapping of the observations made by the robot to the expected or average overall utility of the landing site. Both the observation distribution and expected utility terms are derived from the BN framework in Sec. 4.1.

Optimal decision making in partially observable environments is in general intractable [13]. Previously, we used an approximate sampling solution based on Monte Carlo Tree Search, described in detail in [1]. In our tree search, tree nodes are valid spacecraft states, defined by x-y position, orientation, remaining budget, and altitude. Node edges are the actions the spacecraft can take.

The key idea is to conduct forward simulations in the decision space, sample observations resulting from future movement and sensing actions, and simulate belief updates to estimate rewards. The proposed approach allows principled long horizon planning in an anytime manner while avoiding local minima. Another major advantage of our approach is that it is sensor agnostic and can be applied to arbitrary dynamic models, and number of sensors without algorithmic modifications.

5 COMPLEXITY ANALYSIS

Evaluating the reward of forward simulated candidate trajectories is the most computationally intensive operation in planning. As mentioned in Section 4.2, reward is calculated by sampling observations along candidate trajectories and propagating belief updates to deduce the best landing site. The most expensive term to compute in this process is determining the posterior utility of landing sites given observations, $P(U|Z)$, computed with Eq. 3. We define T_L as the time to compute $P(U|Z)$ for one landing site.

Since we have to evaluate $P(U|Z)$ for each landing site observed during the forward simulation, the complexity of evaluating the reward for a single forward simulation is $O(N_{sites}T_LH)$ where N_{sites} is the number of sites updated for observations taken at a single timestep, and H is the number of actions in the planning horizon. The total time taken for the MCTS is therefore $O(N_S N_{sites} T_L H)$ where N_S is the number of forward simulations.

This complexity analysis shows there are two ways to reduce computation time: reduce the inference time, T_L, or reduce the number of times inference has to be done. The second can be achieved by reducing one of the planning horizon, the number of sites updated during the forward simulation, or the number of forward simulations, N_S, needed to find good trajectories. We limit the discussion to reducing inference times and reducing number of sites updated.

Reducing Inference Time: An analytic computation of $P(U|Z)$ requires summing over all possible instances of the feature space vector, $\mathbf{F}$, as in Eq. (3), which grows rapidly with the size of $\mathbf{F}$. We can construct simpler, faster Bayes nets by mapping sensor data to a smaller, discretized, semantic feature space instead of using raw features. However, semantic features can be expensive to compute. Trade-offs between feature expressiveness, dimensionality, and simplicity is application dependent. For larger and more complex BN's, we can use approximate Bayesian inference techniques [14], but in this problem instance, our combination of particle filter updates and message passing already yields near optimal inference times.

Reducing Number of Sites Updated: Currently, the MCTS samples observations and updates beliefs for every site seen during a forward simulation. The maximum number of sites seen during a single sense for a robot with a circular sensor FoV (Fig. 3) is given by Eq. 6:

$$\begin{aligned} N_{sites} &= FoVArea \times GridResolution \\ &= \pi\left(A \tan\frac{\theta}{2}\right)^2 GridResolution \end{aligned} \quad (6)$$

where A is the lander altitude and θ is the sensor viewing cone angle. N_{sites} is upper bounded by the number of sites on the map. A and θ are system parameters and outside the control of the planning algorithm. We could adapt the grid resolution during the mission. Early in the mission, when the robot is at high altitude, little information is lost by using a coarse grid. As the robot approaches a landing site, we can increase the resolution, similar to the approach of Popovic et al. [15]. The challenge remains in preserving and transforming information through different resolutions.

Alternatively, instead of updating beliefs on all the sites seen during a forward simulation, we can

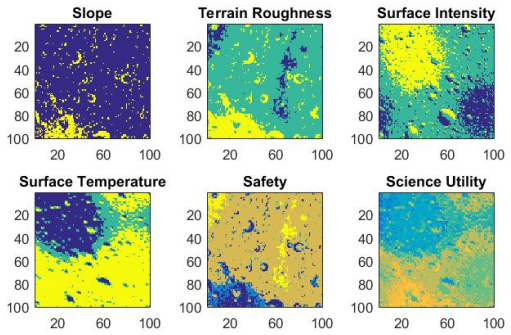

Figure 4: The generated feature maps along with the corresponding safety and science utility

track the beliefs of a subset of sites which are more promising landing sites, a concept known as importance sampling [4]. This will cut the planning time by a ratio equal to $\frac{N_{tracked}}{N_{sites}}$- a significant speedup for large environments. The problem remains of deciding how many sites to track and how to track these sites.

At the beginning of forward simulations, we sample the top $N\%$ of landing sites based on their current expected utility. During forward simulations, we only sample observations and update beliefs of sites in this set. Importance sampling ensures that information gain on only the promising sights is considered and the lander does not waste planning time tracking sites that are likely low quality.

6 EXPERIMENTAL SETUP

Map Generation: Previously [1], our experiments used randomly generated 10×10 and 20×20 feature maps. In this paper, we generate geologically realistic maps. Feature maps are created by procedural generation of rocks, craters, and striae on icy terrain. Rock and crater size and placement are governed by power-law distributions commonly found on airless bodies, while the striae are mostly qualitative. Slope and terrain roughness of local regions, the first two features, were calculated from the terrain map. We used the Stefan-Boltzmann law to estimate surface temperature for a Europa-like body, while the visual appearance of the surface was rendered with the Hapke and Phong reflectance functions and mixed with pre-generated icy textures. Surface temperature and intensity formed the last two features. All features were discretized into three categories– low, medium and high.

We sub-sampled a map of size 100 × 100 which had a large diversity of features. This equates to a total of 10000 sites the robot can potentially land on, and replicates the large scale nature of real world missions. Ground truth safety of the sites was set to be a function of the slope and terrain roughness of the site given by Table 1.

Table 1: Landing site safety score as a function of discretized terrain geometry features.

		Slope		
		L	M	H
Terrain Roughness	L	1	0.8	0.4
	M	0.8	0.6	0.2
	H	0.4	0.2	0

The science utility of a landing site x was defined as a weighted function of all four site features, Slope, Terrain Roughness, Surface Intensity, and Surface Temperature:

$$S(x) = 0.1F_S + 0.2F_{TR} + 0.3F_{SI} + 0.5F_{ST} \quad (7)$$

Applying Table 1 and Eq. (7) to the feature maps yields the maps shown in Fig. 4. The overall utility is defined as the product of safety and science utility. The weights as well as which features affect safety and science utility are arbitrarily set and can be varied as per mission requirements.

Lander Parameters: The simulated lander is equipped with two sensors: a visual sensor that takes noisy observations of slope and roughness, and a spectrometer that can take noisy measurements of temperature and intensity. Both sensors have noise models discussed earlier in Eq. 4 with a circular field of view with a viewing cone angle of 8 degrees and $RMax$ of 100 units. The visual sensor has a maximum accuracy of 80% while the spectrometer has a maximum accuracy of 95%. G_p is set to be a uniform distribution.

For illustration we simplified the planning problem to the x-y domain where the descent rate and speed of the lander are fixed at 1.5 and 5 units per time step. The lander motion primitives were chosen to be Dubin's curves which orientate the lander in -45, -30, 0, 30 and 45 degrees relative to the current orientation. Since there were two sensors, and in each time step the lander can choose a motion primitive and type of sensor to use, the total action space is of size 10. The cost for using the visual sensor was 1 unit while the cost for using the spectrometer was 5 units.

Algorithms: To study the effect of importance sampling on planning time and landing site quality we compared with approximation factors of

40%, 20%, 10%, 5%. We compare against the algorithm from [1], labeled 'original', a random planning baseline where random actions are taken, observations are collected, and landing site is the best site within the landing radius of the terminal state, and a greedy baseline which selects and lands at the best site in the orbital data.

20 trials were run with 5 trials starting from the center of each of the four edges of the map with an initial altitude of 50 units. The lander was given a sensing budget of 50 units. Orbital data was generated by convolving a 9×9 median filter over the ground truth feature maps to the create low resolution data. The Bayesian priors on landing site features were initialized with the orbital data.

7 RESULTS

The utility of the algorithms' selected landing sites is shown in Fig. 5. 'Global' gives the true utility of all landing sites in the map. 'Random' yields higher utility landing sites than 'global', showing the advantage of using collected observations. Our MCTS-based approaches and and the greedy algorithm have the same median utility. However, the sites selected by greedy have greater variability, which often leads to sites with low utility. This occurs when low quality orbital data masks the true value of the sites. If the orbital data is poor enough, the performance of the greedy approach could worsen arbitrarily.

The median utility of sites is unchanged as the proportion of sites tracked, N, is decreased. Further, as N is reduced, the tracked sites had a tendency to be close to each other, since nearby sites tend to have similar features, as seen in Fig. 4. As a result the spacecraft often flew towards the same set of sites, which could lead to being stuck in local minima. Importance sampling strategies that incorporate spatial variability and site utility uncertainty into the sampling process instead of only expected utility remains for future investigation.

The planning times are given in Table 2. Minimum times occur near the end of the mission, when the budget or altitude are near zero. Maximum planning times occur near the beginning of missions when the robot forward simulates trajectories for the entire horizon. The minimum time remains constant, while maximum significantly reduces as N as decreased. At $N = 5\%$, the difference between minimum and maximum times is 4 seconds, while in the full MCTS the difference is 40.6 seconds. Importance sampling mitigates the effects of planning horizons on planning time meaning we can search even longer horizons.

When horizons are near zero, constant time operations dominate planning time. Further analysis showed that is was largely due to the copying of the probability distributions of all landing sites before each forward simulation. More efficient data structures that only copy the relevant parts of the belief space should lead to significant reductions in planning times.

Table 2: Minimum and maximum run time for 200 iterations of MCTS

	Min time (s)	Max time (s)
Original	27.6	68.2
40%	27.7	54.8
20%	27.7	41.5
10%	27.4	34.7
5%	27.3	31.3

8 CONCLUSIONS

This paper discussed a sampling based descent trajectory planner that enables spacecraft to exploit information gained during descent to select landing sites that are both safe and have high science utility. Our approach reduces precursor data quality requirements, potentially reducing costs of both icy moon missions and robotic lander missions in general. We derived the complexity of our approach and used importance sampling to improve scalability.

Our modifications substantially reduced the planning time without affecting the quality of landing sites. In future work, we would like to experiment with reward functions that do not require sampling observations and simulating belief updates (and hence much faster to evaluate), and integrate our approach with continuous space optimizing techniques to achieve an end to end solution for EDL.

Acknowledgements

This work was supported by the ICICLES project as part of NASA's Coldtech grant.

References

[1] Arora A, Furlong M, Wong U, Sukkarieh S and Fong TW (2017) Multi-modal Active Perception for Autonomously Selecting

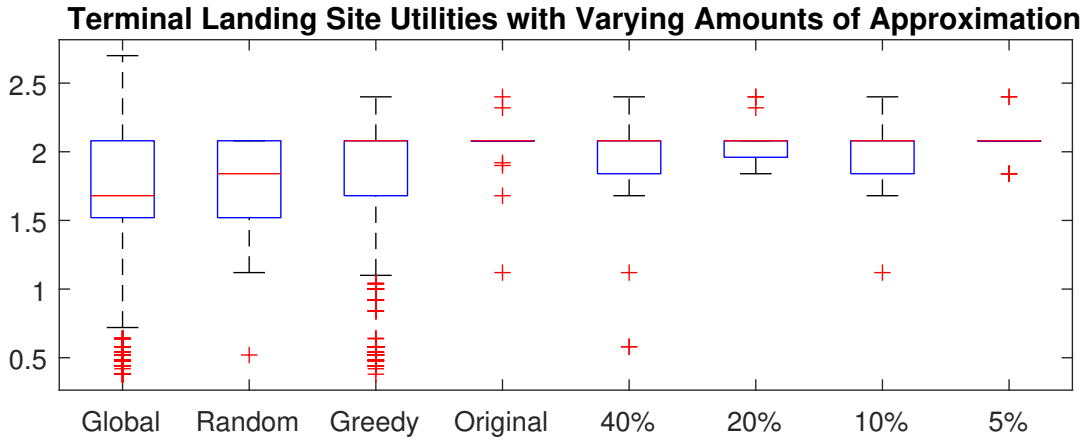

Figure 5: Distribution of Utility of landing sites achieved by the different algorithms. Both the Greedy and MCTS-based algorithms produce the same median score, but the MCTS-based algorithms reduce the variance in utility, relative to the Greedy.

Landing Sites on Icy Moons. In: *Proc. of AIAA Space Forum and Exposition*, p.5182.

[2] Board SS, Council NR et al. (2012) *Vision and voyages for planetary science in the decade 2013-2022*. National Academies Press.

[3] Browne CB, Powley E, Whitehouse D, Lucas SM, Cowling PI, Rohlfshagen P, Tavener S, Perez D, Samothrakis S and Colton S (2012) A survey of monte carlo tree search methods. In: *IEEE Trans. Comp. Intel. AI*, 4(1):pp.1–43.

[4] Glynn PW and Iglehart DL (1989) Importance sampling for stochastic simulations. In: *Management Science*, 35(11):pp.1367–1392.

[5] Johnson A and Ivanov T (2011) Analysis and testing of a lidar-based approach to terrain relative navigation for precise lunar landing. In: *Proc. of AIAA-GNC*.

[6] Ivanov T, Huertas A and Carson J (2013) Probabilistic Hazard Detection for Autonomous Safe Landing. In: *Proc. of AIAA-GNC*.

[7] Serrano N (2006) A bayesian framework for landing site selection during autonomous spacecraft descent. In: *Proc. of IEEE/RSJ IROS*, IEEE, pp.5112–5117.

[8] Desaraju VR, Michael N, Humenberger M, Brockers R, Weiss S and Matthies LH (2014) Vision-based Landing Site Evaluation and Trajectory Generation Toward Rooftop Landing. In: *Proc. of RSS*.

[9] Shen Z and Lu P (2003) Onboard generation of three-dimensional constrained entry trajectories. In: *Journal of Guidance, control, and Dynamics*, 26(1):pp.111–121.

[10] Arora A, Fitch R and Sukkarieh S (2017) An Approach to Autonomous Science by Modeling Geological Knowledge in a Bayesian Framework. In: *Proc. of IEEE/RSJ IROS*.

[11] Fitzgerald D, Walker R and Campbell D (2005) A vision based forced landing site selection system for an autonomous UAV. In: *Proc. of ISSNIP*, IEEE, pp.397–402.

[12] Gupta RP (2017) *Remote sensing geology*. Springer.

[13] Cassandra AR (1998) *Exact and approximate algorithms for partially observable Markov decision processes*. Brown University.

[14] Nielsen TD and Jensen FV (2009) *Bayesian networks and decision graphs*. Springer Science & Business Media.

[15] Popovic M, Vidal-Calleja T, Hitz G, Sa I, Siegwart R and Nieto J (2017) Multiresolution mapping and informative path planning for UAV-based terrain monitoring. In: *Proc. of IEEE/RSJ IROS*.