

Energy-Constrained Active Exploration Under Incremental-Resolution Symbolic Perception

Disha Kamale¹, Sofie Haesaert², Cristian-Ioan Vasile¹

Abstract—In this work, we consider the problem of autonomous exploration in search of targets while respecting a fixed energy budget. The robot is equipped with an *incremental-resolution symbolic perception* module wherein the perception of targets in the environment improves as the robot’s distance from targets decreases. We assume no prior information about the total number of targets, their locations as well as their possible distribution within the environment. This work proposes a novel decision-making framework for the resulting constrained sequential decision-making problem by first converting it into a reward maximization problem on a product graph computed offline. It is then solved online as a Mixed-Integer Linear Program (MILP) where the knowledge about the environment is updated at each step, combining automata-based and MILP-based techniques. We demonstrate the efficacy of our approach with the help of a case study and present empirical evaluation in terms of expected regret. Furthermore, the runtime performance shows that online planning can be efficiently performed for moderately-sized grid environments.

I. INTRODUCTION

Robotic exploration for critical missions such as post-disaster search-and-rescue (SaR), extra-terrestrial exploration for scientific data collection, etc. demand promising and time-optimal solutions. For exploration in search of objects of interest (called *targets*), an ideal approach would entail visiting all regions of the environment. However, in practice, these problems are resource-constrained due to computationally expensive information acquisition and hardware limitations such as finite battery life.

The problem is made further challenging by the limited information that is available a-priori, for instance, knowing the precise locations of victims to be rescued during an SaR mission. This amplifies the difficulty in carrying out a well-informed search and necessitates strategic use of robot’s observation history to direct the search to regions with high likelihood of containing targets [1]–[4] or the estimation the risk of misperception [5] for safe exploration. In this work, we consider the robot perception model that provides partial symbolic information incrementally as the distance to yet to explore regions decreases as first proposed in [6]. For instance, consider an autonomous robot with a fixed energy budget deployed in a SaR environment (Fig. 1) where it is required to extinguish as many instances of fire and rescue as many victims as possible before reaching the EXIT. The

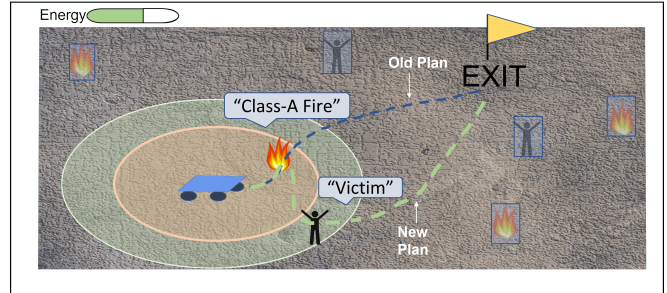


Fig. 1: An autonomous robot is required to service as many targets (Fire, Victim) as possible, and finally reach the “EXIT” before its energy runs out. Onboard is a robot-centric incremental-resolution perception module having limited sensing range (green boundary). The robot can exactly observe the orange region and can observe some limited information about the green region. Other instances of Fire and Victim are not visible to the robot.

robot can exactly observe the orange region and some limited information about the surrounding green region. This gives rise to a constrained sequential decision-making problem.

The problem, in its general form, is modeled as a Partially-Observable Markov Decision Process (POMDP) [2], [7]–[9] or a belief-space Markov Decision Process (MDP) [10]. Several approaches have been proposed for planning in belief spaces [11], [12]. Nevertheless, these formalisms have a limited scalability with the size of the environment. In [13], the authors present the Next Best View online planning approach that optimizes the selection of the most informative viewpoint to obtain information about the unobserved space. Similarly, the frontier-based exploration approaches aim to traverse large frontiers in order to maximize the knowledge about the environment [14].

The resource constraints inherently give rise to an exploration-exploitation trade-off which is well-studied in the reinforcement learning literature [1], [15], [16]. However, training for these approaches is often time and resource-intensive. Many existing works employ sampling-based approaches for multi-objective exploration [17], reward shaping [16], reactive service of local dynamic requests [18]. On the other hand, several automata-based [19]–[21] and optimization-based [22], [23] approaches have been proposed for satisfaction of complex tasks. [22], [23] consider Mixed-Integer Linear Programming (MILP) techniques. Although MILPs are in general NP-hard, efficient off-the-shelf tools exist that facilitate real-time control synthesis [24].

This work addresses the problem of energy-constrained exploration in search of targets whose locations, numbers are a-priori unknown. While exploring, the robot observes and tracks the symbolic label associated with each grid cell in

¹Disha Kamale and Cristian-Ioan Vasile are with the Mechanical Engineering and Mechanics Department, Lehigh University, Bethlehem, PA 18015 {ddk320, cvasile}@lehigh.edu

²Sofie Haesaert is with the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, Netherlands S.Haesaert@tue.nl

This work is supported by the Dutch NWO Veni project CODEC (project number 18244)

an incremental-resolution manner. This differs from existing works in several aspects. Specifically, as opposed to [20], we consider a static, incrementally-sensed rewards to be maximized on a fixed energy budget. As opposed to [7], [25], [26] no information about the numbers, locations and possible distributions of the targets is assumed. In [14] and [27], fixed budget exploration to maximize the information about the environment. The goal our work is to maximize collection of specific targets whose locations are a-priori unknown.

In this work, we propose a decision-theoretic framework in which the product graph between the robot motion and the energy available for motion is pre-computed. Using this, an online planning algorithm that solves a Mixed-Integer Linear Program at each time step by utilizing the observations given by an incremental-resolution symbolic perception module. Our approach preserves the accumulation of symbolic information as the robot moves through the environment and turns the problem of constrained planning for maximizing target collection from unknown locations into a deterministic optimal flow problem with respect to the current knowledge about the environment.

The main contributions of this work are: 1) Building on the framework in [6], we propose the problem of energy-constrained exploration with incremental-resolution symbolic perception where no information about the targets is assumed. 2) We present abstraction models that formally capture this problem and propose a decision-theoretic framework that combines randomized allocation of energy with automata-based and MILP-based approaches. 3) We highlight the performance of the proposed planning framework using case studies and empirically evaluate the performance in terms of expected regret. Additionally, we characterize the performance of the online planning algorithm using the runtime performance.

Notation: A set of integers starting at a and ending at b , both inclusive, is denoted by $[[a, b]]$. The 1-norm of a vector x is denoted as $\|x\|_1$. The sets of integers and non-negative integers are denoted as \mathbb{Z} and $\mathbb{Z}_{\geq 0}$. Let $\mathbb{B} = \{0, 1\}$. $\mathbf{1}_{f=a}$ denotes the indicator function which is 1 if $f = a$ and 0 otherwise.

II. PROBLEM SETUP

Consider a planar grid environment with objects of interest (e.g., artifacts, victims), referred to as **targets**, see Fig. 2. The targets are hidden in the environment and no prior information about the total number of targets, their locations or the frequencies of their occurrences is available. Given an autonomous robot with a **fixed energy budget**, our objective is to design a path that services as many of these targets as possible and ultimately reaches the given **goal location**. Next, we describe various components of the problem.

Robot Motion and Environment. Let X denote the set of locations, i.e., grid cells. The robot can move *North*, *East*, *South*, *West* to adjacent locations deterministically. The initial and goal locations, x_{init} and x_{goal} , are given.

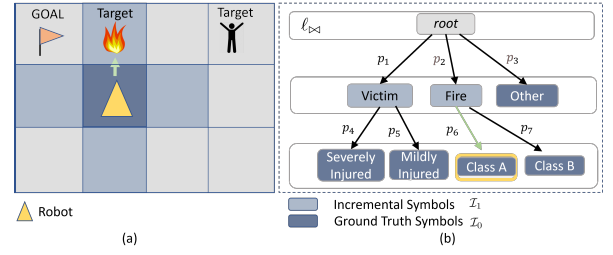


Fig. 2: (a) A robot with a sensing range limited to its immediate neighbors (blue cells), and has no observations about all grey cells. (b) shows an instance of refinement in symbolic perception where the *root* indicates no observation is available. For each subsequent layer, the symbols are refined. Thus, (a) and (b) depict the metric and symbolic representation of the agent's perception, respectively. As the robot moves along the green arrow, its observation about cell x_1 changes from Fire to Class_A.Fire with probability p_6 .

Let the set of *targets* be denoted by \mathcal{L} . Each location may contain a single target, another object of no interest to the robot or nothing at all. Let Π contain all symbols of targets and objects. Let \mathcal{I}_0 denote a set of ground truth symbols that correspond to full semantic information, $\mathcal{I}_0 \subset \Pi$. We associate each cell with a *ground truth symbol* from \mathcal{I}_0 that includes the targets, $\mathcal{L} \subseteq \mathcal{I}_0$. A cell that does not contain targets or objects is called *empty* associated with the symbol $\ell_0 \in \Pi$. We assume that the environment is static and the ground truth symbols are independent between locations.

This work focuses on the high-level motion of the robot in the grid environment. We assume the low-level controllers to enforce the motion of the robots in the grid are available.

Incremental-resolution Perception. The robot is equipped with a perception module necessary for detecting the potential targets. We consider sensing within a limited range around the robot that provides observations with *symbolic resolution* decreasing with the distance from the robot. The sensing range $D \in \mathbb{N}$ allows the robot to observe all cells x' within D -Manhattan distance away from the current cell x , i.e., $\|x - x'\|_1 \leq D$ for all $x' \in X$. We denote by \mathcal{N}_x^d all cells x' for which $\|x' - x\|_1 = d$, where x is the current robot location and $d \in \mathbb{Z}_{\geq 0}$. Thus, the set of visible cells is $\mathcal{N}_x^{\leq D} = \mathcal{N}_x^0 \cup \dots \cup \mathcal{N}_x^D$.

The symbolic perception information is modeled in an incremental-resolution manner as described in [6]. As the robot moves through the environment, it deterministically observes the symbols of cells within the sensing range D at different resolution levels depending on its distance from them. For each distance $d \in \mathbb{Z}_{\geq 0}$, we associate a set of symbols \mathcal{I}_d that can be observed at the Manhattan distance d by the robot. The set of all symbols is denoted by $\mathcal{I} = \bigcup_{d=0}^D \mathcal{I}_d$. At its current location, the robot can observe only ground truth symbols, i.e., $\mathcal{I}_0 \subseteq \Pi$. Some ground truth symbols may be observed from farther away ($d \geq 1$), see Fig. 2. The symbols $\mathcal{I} \setminus \Pi$ are called *incremental symbols* and capture lower resolution (incomplete) semantic information.

Moreover, we are given a priori distribution on what symbols may be observed by moving one cell closer to an observed cell x' . Formally, for any $\ell \in \mathcal{I}_d$ the prior probability distribution $p_\ell : \mathcal{I}_{d-1} \rightarrow [0, 1]$ is given, where $p_\ell(\ell') > 0$ iff the symbol ℓ' can be observed for cell x' .

at distance $d - 1$ given that at distance d symbol ℓ was observed. For distance $d \geq D + 1$, no observations are available. For uniformity of presentation, we associate this mode with a *root* symbol ℓ_{\boxtimes} and prior $p_{\ell_{\boxtimes}} : \mathcal{I}_D \rightarrow [0, 1]$, where $p_{\ell_{\boxtimes}}(\ell) > 0$ for all $\ell \in \mathcal{I}_D$. The relationship captured by priors $\{p_{\ell}\}_{\ell \in \mathcal{I} \cup \{\ell_{\boxtimes}\}}$ represents the *perception refinement* structure of the robot's sensing.

Example II.1. Consider the robot in the 4×4 grid environment in Fig. 2(a) containing fire and victim targets. The robot needs to perform as many instances of rescuing severely injured victims and extinguishing Class-A fire as possible, if they are present, and reach the goal location. The set of symbols is $\mathcal{I}_0 = \{\text{Severely_Injured_Victim, Mildly_Injured_Victim, Class_A_Fire, Class_B_Fire, Other}\}$; $\mathcal{L} = \{\text{Severely_Injured_Victim, Class_A_Fire}\}$.

Fig. 2(b) is an instance of *perception refinement* with $D = 1$. The directed edges indicate the evolution of symbolic information. The robot can observe the ground truth symbols \mathcal{I}_0 for its current location and the incremental symbols $\mathcal{I}_1 = \{\text{Victim, Fire, Other}\}$ for the cells that are one-step away. The root symbol ℓ_{\boxtimes} corresponds to no observation available and that is the current robot observation for cells beyond sensing range. Thus, the (a) and (b) denote the metric and symbolic representation of the perception model.

Energy Constraints and Target rewards. The robot has a fixed energy budget \mathbf{E} that implicitly bounds the horizon over which the exploration mission can be performed. Each transition between adjacent grid cells $x, x' \in X$ takes $w(x, x') > 0$ energy.

A target in the environment is *satisfied* when the robot moves to a cell containing that target and collects a *target reward* $\mathbf{r}(\ell)$, $\ell \in \mathcal{L}$. The energy required for servicing each symbol is captured by the map $\mathbf{e} : \Pi \rightarrow \mathbb{Z}_{\geq 0}$. For all targets $\ell \in \mathcal{L}$ the servicing energy $\mathbf{e}(\ell) > 0$, while for any other symbol $\mathbf{e}(\ell) = 0$, $\forall \ell \in \Pi \setminus \mathcal{L}$.

Problem II.1 (Energy-Constrained Active Exploration). Given a robot with incremental-resolution symbolic perception refinement $\{p_{\ell}\}_{\ell \in \mathcal{I} \cup \{\ell_{\boxtimes}\}}$, the energy budget \mathbf{E} deployed in a grid environment with unknown cell symbols, and the set of ground truth symbols \mathcal{L} with servicing rewards \mathbf{r} and energy costs \mathbf{e} , find a path such that the robot maximizes the sum of target servicing rewards and reaches the goal location x_{goal} while respecting the energy budget.

III. APPROACH : ABSTRACTION MODELS

This section presents the formal representations of robot motion and perception abstractions that represent the components described in Section II.

A. Robot Motion and Environment

Definition 1. A weighted transition system (TS) is a tuple $\mathcal{T} = (X, x_0^T, \delta_{\mathcal{T}}, \Pi, h, w)$, where: X is a finite set of states associated with the grid's cells; $x_0^T = x_{\text{init}} \in X$ is the initial state; $\delta_{\mathcal{T}} \subseteq X \times X$ is a set of transitions; Π is a set of symbols (atomic propositions); $h : X \rightarrow \Pi$ is a labeling function; and $w_{\mathcal{T}} : \delta_{\mathcal{T}} \rightarrow \mathbb{Z}_{>0}$ is a weight function.

The robot motion in the environment is abstracted as a weighted transition system \mathcal{T} . The state transition function

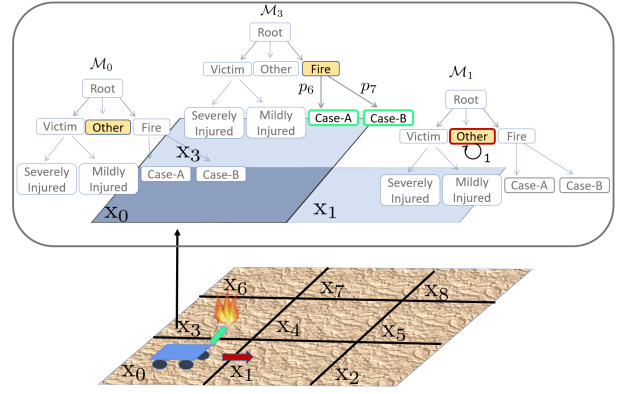


Fig. 3: The symbolic tracking models for cells x_0, x_1 and x_3 considering $d = 1$. The rest of the grid cells are yet to be observed.

$\delta_{\mathcal{T}}$ is deterministic. The state space X constitutes the cells of the grid environment in which the robot performs assigned mission. The set of atomic propositions comprises of all ground truth symbols Π . Note that, as opposed to the standard setting, the labeling function $h(\cdot)$ in Def. 1 is unknown. As the robot can only sense the environment locally, $h(\cdot)$ is locally observed during deployment.

We define a *path* of the system as a finite sequence of states $\mathbf{x} = x_0 x_1 \dots x_m$ such that $(x_k, x_{k+1}) \in \delta_{\mathcal{T}}$ for all $k \geq 0$, and $x_0 = x_0^T$. The set of all trajectories of \mathcal{T} is $\text{Runs}(\mathcal{T})$. We define the weight of a trajectory as $w_{\mathcal{T}}(\mathbf{x}) = \sum_{k=1}^{|\mathbf{x}|} w_{\mathcal{T}}(x_{k-1}, x_k)$. The set of states visited by trajectory \mathbf{x} is $\text{Vis}(\mathbf{x}) = \{x \mid \exists k \in [0, |\mathbf{x}|] \text{ s.t. } x_k = x\}$.

B. Incremental-resolution Symbolic Tracking Model

Given a past observation ℓ of cell x' at distance d , only the action of moving closer to x' leads to an observation ℓ' of higher resolution. Since the environment is static, and the robot makes observations deterministically, the robot's *knowledge* about cell's symbols is cumulative, i.e., it does not forget prior observations. Thus, the evolution in the robot's knowledge about the environment is governed not only by the probabilities of the perception refinement, but also by its evolution of distances from cells throughout the mission. We capture the robot's knowledge about the environment as Markov Decision Processes (MDP) that we refer to as *symbolic tracking model*.

Definition 2 (Markov Decision Process). A finite, stationary discrete-time Markov Decision Process (MDP) is a tuple $\mathcal{M} = (M, m_0, \mathbb{U}, \mathbb{P})$, where M is the state space, m_0 denotes the initial state, \mathbb{U} is the input space, and $\mathbb{P} : M \times \mathbb{U} \times M \rightarrow [0, 1]$ is a next-state transition probability function such that for all states $m \in M$ and inputs $u \in \mathbb{U}$, we have $\sum_{m' \in M} \mathbb{P}(m, u, m') \in \mathbb{B}$.

A trajectory of \mathcal{M} is a sequence of states and inputs starting in state m_0 denoted as $m_0, u_0, m_1, u_1, m_2, u_2, \dots$ where u_k denotes the input at state x_k .

The state space M corresponds to all possible symbolic observations interpreted as the *knowledge* about a cell x' . Thus, $M = \mathcal{I} \cup \{\ell_{\boxtimes}\}$. The initial state m_0 is the root ℓ_{\boxtimes} indicating that no observation is yet available. The input space $\mathbb{U} = \mathbb{Z}_{\geq 0}$ captures the distances to the observed cell

x' from the robot locations x . The transition probability function maps the current robot's knowledge state m about cell x' and input distance $u = \|x - x'\|_1$ into the next knowledge state m' . Formally, we have

$$\mathbb{P}(m, u, m') = \begin{cases} p_m(m') & m \in \mathcal{I}_{u+1}, m' \in \mathcal{I}_u \\ 1 & m' = m, m \in \mathcal{I}_d, u \geq d \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The transition function in (1) induces a Directed Acyclic Graph (DAG) over the state space M (without considering self-loops). The structure naturally captures the cumulative nature of the symbolic knowledge about a cell x' .

In the following, we denote explicitly the MDP associated with a cell $x' \in X$ by $\mathcal{M}_{x'}$.

Example III.1 (Symbolic tracking model). *Consider Fig. 3. The current MDP states are $m(x_0) = \text{Other}$, $m(x_1) = \text{Other}$, $m(x_3) = \text{Fire}$. If the robot moves to x_3 following the green arrow, the resulting MDP states will be $m(x_3) = \text{Case_A}$ with probability p_6 and $m(x_3) = \text{Case_B}$ with probability p_7 whereas $m(x_1)$ and $m(x_0)$ continue to stay at Other. If, instead, the robot moves to x_1 , the resulting MDP states will be $m(x_1) = \text{Other}$ with probability 1, $m(x_3) = \text{Fire}$, $m(x_0) = \text{Other}$.*

Having established the abstraction models, we now proceed to the decision-making framework. Given \mathcal{T} , we define rewards for all cells that encourage exploration of the environment for targets thereby converting Problem II.1 into a constrained reward maximization problem over \mathcal{T} . The problem is solved at each time step of the mission as the robot observes the environment and its knowledge about cells is updated. We evaluate the efficiency of the reward scheme and reward maximization algorithm via expected regret between the target servicing reward without any prior information vs with full location and symbol information.

$$\begin{aligned} \max_{\mathbf{x} \in \text{Runs}(\mathcal{T})} & R_{\text{path}}(\mathbf{x}) \\ \text{s.t. } & x_0 = x_0^\mathcal{T} \\ & x_T = x_{\text{goal}}, T = |\mathbf{x}| \\ & w_{\mathcal{T}}(\mathbf{x}) + \sum_{x \in \text{Vis}(\mathbf{x})} e(h(x)) \leq \mathbf{E} \end{aligned} \quad (2)$$

where $R(\mathbf{x})$ is a reward function over paths in \mathcal{T} (see Sec. IV-C) defined based on target service rewards \mathbf{r} and robot's knowledge $\mathcal{M}_{x'}$ about cells $x' \in X$ at current time. The rewards in the equation above are path-dependent which is difficult to compute. The proposed approach transforms the objective into a state-dependent reward maximization problem that can be easily solved as an integer linear program. The reward and energy path constraints of the problem are accounted for by the constraints of the MILP and automata construction, respectively.

IV. ALGORITHMS

This section elaborates our approach to Pb. II.1 that aims to find a good solution tractably rather than an optimal one.

Our proposed Energy-constrained Incremental Symbolic Perception (EISP) planning framework proceeds by first decomposing the problem into an offline product space

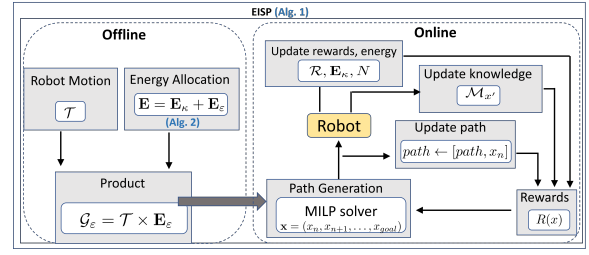


Fig. 4: Components of the proposed EISP planning framework. The product graph computed offline is used for solving an MILP at each step as the robot knowledge about the environment is updated.

Algorithm 1: EISP Planning Algorithm()

Input: $\mathcal{T}, x_{\text{goal}}, \mathcal{L}, D, \mathbf{e}, \mathbf{E}$
Output: path

```

// Offline
1  $\mathbf{E}_k, \mathbf{E}_e, N \leftarrow \text{sample\_energy\_budget}()$  // Alg. 2
2  $\mathcal{G}_e \leftarrow \text{construct\_product\_DAG}()$  // Alg. 3
3  $\text{path} \leftarrow [], x \leftarrow x_0^\mathcal{T}, \mathcal{R}_{\text{serv}} \leftarrow 0$  // Initialize
4 while  $x \neq x_{\text{goal}}$  do // Online
5    $\text{path} \leftarrow [\text{path}, x_c]$  // Update path
6   Get observations  $\ell_x(x')$  at  $x$  for all  $x' \in \mathcal{N}_x^{\leq D} - \text{Sec. II}$ 
7   Update  $\mathcal{M}_{x'}$  with  $u = \|x - x'\|_1$ , for all  $x' \in \mathcal{N}_x^{\leq D} - \text{Sec. III-B}$ 
8   if  $e(h(x)) \leq \mathbf{E}_k$  then // Target servicing updates
9      $\mathbf{E}_k \leftarrow \mathbf{E}_k - e(h(x))$ 
10     $\mathcal{R}_{\text{serv}} \leftarrow \mathcal{R}_{\text{serv}} + \mathbf{r}(h(x))$  // Servicing rewards
11  else No service
12    Update rewards  $R(\cdot) - \text{Sec. IV-C}$ 
13     $x_{\text{next}} \leftarrow \text{solve\_milp}(\mathcal{G}_e, x, R(\cdot)) - \text{Sec. IV-D}$ 
14     $\mathbf{E}_e \leftarrow \mathbf{E}_e - w_{\mathcal{T}}(x, x_{\text{next}})$  // Update motion energy
15     $x \leftarrow x_{\text{next}}$  // Robot moves to  $x_{\text{next}}$ 
16 return path

```

construction and an online planning problem. Given the energy constraints, the offline algorithm heuristically estimates the energy required for satisfying potential targets thereby splitting the available energy budget into the energy for target satisfaction and for exploration (Alg. 2). The reachable solution space is then pre-computed based on the energy allocation for exploration and the robot motion model (Alg. 3). Finally, a modified optimal flow problem is solved online at each step on the pre-computed product graph (Alg. 1). Specifically, the observations of the cells within the sensing range are used to update rewards and the energy available for servicing targets in case a target is serviced. With the updated rewards, solving the MILP (5) determines the maximum reward path to the goal x_{goal} . The first step is executed and the process is repeated until the robot reaches x_{goal} . An outline of EISP planning algorithm is shown in Fig. 4. In what follows, we present a detailed discussion about each of these components.

A. Energy allocation for servicing targets and exploration

Since the targets' number, locations and occurrences of each type are a priori unknown, the robot must decide how to allocate its energy budget \mathbf{E} for exploring versus collecting targets it might find with energy costs \mathbf{e} . In our approach outlined in Alg. 2, the energy budget is divided offline into the energy for collecting targets \mathbf{E}_k and that for exploration \mathbf{E}_e . First, we compute the energy required to ensure the robot reaches the goal x_{goal} . Next, we draw the frequencies of occurrence of each target denoted as α_ℓ from some arbitrary distribution \mathcal{D} (line 4) until the target collection energy \mathbf{E}_k

Algorithm 2: sample_energy_budget()

Data: $\mathbf{E}, \mathbf{e}, \mathcal{L}, x_0^T, x_{goal}$
Result: $\mathbf{E}_\kappa, \mathbf{E}_\varepsilon, N$

- 1 Initialize $\mathbf{E}_\kappa = \mathbf{E}$
- 2 $\mathbf{E}_{goal} = \|x_{goal} - x_0^T\|_1$ // Min energy to goal
- 3 **while** $\mathbf{E}_\kappa \geq \mathbf{E} - \mathbf{E}_{goal}$ **do**
- 4 Draw $\{\alpha_\ell\}_{\ell \in \mathcal{L}} \sim \mathcal{D}(\cdot)$, $\alpha_\ell \in \mathbb{Z}_{\geq 0} \forall \ell \in \mathcal{L}$
- 5 $\mathbf{E}_\kappa = \sum_{\ell \in \mathcal{L}} \alpha_\ell \cdot \mathbf{e}(\ell)$ // Energy estimate for targets
- 6 **return** $\mathbf{E}_\kappa, \mathbf{E}_\varepsilon = \mathbf{E} - \mathbf{E}_\kappa, N = \mathbf{E}_\kappa / \min_{j \in \mathcal{L}} \{\mathbf{e}_j\}$

(line 5) is less than the available energy $\mathbf{E} - \mathbf{E}_{goal}$. Finally, we compute the upper bound on the number of targets to be collected by dividing \mathbf{E}_κ by the minimum energy to service a target (line 6).

Proposition 1. *The number of targets that robot services is upper bounded by $N = \mathbf{E} / \min_{j \in \mathcal{L}} \{\mathbf{e}_j\}$, see Alg. 2.*

B. Product Graph Construction

As the energy \mathbf{E}_ε allocated by Alg. 2 may only deplete as the robot moves through the grid, it provides a "directionality" to the planning problem. Leveraging this, we construct a product graph between the robot motion model and an enumeration of \mathbf{E}_ε resulting in an DAG defined as follows.

Definition 3 (Product Graph). *Given the robot motion model $\mathcal{T} = (X, x_0^T, \delta_\mathcal{T}, \Pi, h)$ and the energy available for exploration \mathbf{E}_ε , the product graph is a tuple $\mathcal{G}_\varepsilon = (V_\varepsilon, v_0^\varepsilon, \Xi_\varepsilon, F_\varepsilon)$, where $V_\varepsilon \subseteq X \times [[0, \mathbf{E}_\varepsilon]]$ is the state space, $v_0^\varepsilon = (x_0, \mathbf{E}_\varepsilon)$ denotes the initial state, $\Xi_\varepsilon \subseteq V_\varepsilon \times V_\varepsilon$ represents the transition function, and F_ε is a set of final states*

The transition $((x, e), (x', e'))$ if and only if $(x, x') \in \delta_\mathcal{T}$ and $e' = e - w_\mathcal{T}(x, x')$. The product graph synchronously captures the robot's motion and energy constraints.

Alg. 3 outlines the procedure for computing \mathcal{G}_ε . Starting at the initial state $(x_0^T, \mathbf{E}_\varepsilon)$ (line 1), each transition in \mathcal{T} is iteratively considered and the energy available at the grid cell is updated based on the transition energy cost (line 6). The product model is updated (lines 9-13) and the process continues until the available energy is consumed.

Algorithm 3: construct_product_DAG()

Data: $x_0^T, x_{goal}, \mathbf{E}_\varepsilon = \mathbf{E} - \mathbf{E}_\kappa, \mathcal{T}$
Result: $\mathcal{G}_\varepsilon = (V_\varepsilon, v_0, \Xi_\varepsilon, F_\varepsilon)$

- 1 $V_\varepsilon \leftarrow \{(x_0^T, \mathbf{E}_\varepsilon)\}, \Xi_\varepsilon \leftarrow \emptyset, F_\varepsilon \leftarrow \emptyset$ // Initialize
- 2 $stack \leftarrow V_\varepsilon$
- 3 **while** $stack \neq \emptyset$ **do**
- 4 $(x, e) \leftarrow stack.pop()$
- 5 **forall** $(x, x') \in \delta_\mathcal{T}$ **do**
- 6 $e' \leftarrow e - w_\mathcal{T}(x, x')$
- 7 **if** $e' \geq 0$ **then**
- 8 **if** $(x', e') \notin V_\varepsilon$ **then**
- 9 $V_\varepsilon \leftarrow V_\varepsilon \cup (x', e')$ // Add nodes
- 10 $stack.push((x', e'))$
- 11 **if** $x' = x_{goal}$ **then** $F_\varepsilon \leftarrow F_\varepsilon \cup \{(x', e')\}$
- 12 $\Xi_\varepsilon \leftarrow \Xi_\varepsilon \cup ((x, e), (x', e'))$ // Add edges
- 13 **return** \mathcal{G}_ε

C. Online Planning

Given the pre-computed \mathcal{G}_ε , the robot utilizes the observations made at runtime to synthesize a path from its current cell to the goal that may lead to targets. To incentivize exploration for targets, we introduce rewards as follows.

1) **Reward Design for Active Exploration:** The expected target reward for cell x with MDP state $m(x)$ in \mathcal{M}_x is

$$\mathbb{E}_{\mathcal{M}_x}[r(\mathcal{L}) \mid m(x)] = \sum_{\ell \in \mathcal{L}_{m(x)}} \mathbb{P}(\ell \mid m(x)) \cdot \mathbf{r}(\ell) \quad (3)$$

where $\mathcal{L}_{m(x)} = \{\ell \in \mathcal{L} \mid \ell \preceq m(x)\}$ is the set of targets that may be observed given $m(x)$, and \preceq is the descendent relation in DAG \mathcal{M}_x . The probabilities in (3) are given by

$$\mathbb{P}(\ell \mid m(x)) = \sum_{br \in \mathfrak{P}_{\mathcal{M}_x}^{m(x), \ell}} \prod_{(\ell^{pa}, \ell^n) \in br} p_{\ell^{pa}}(\ell^n)$$

where $\mathfrak{P}_{\mathcal{M}_x}^{m(x), \ell}$ is the finite set of all directed paths from $m(x)$ to ℓ in DAG \mathcal{M}_x , and $p_{\ell^d} : \mathcal{I}_d \rightarrow [0, 1]$, $d \in [[0, D]]$, are the a priori distributions, see Sec. II.

For an observed cell x ($m(x) \neq \ell_{\boxtimes}$), we assign the expected target reward (3) if it was not previously visited and it may still contain targets ($\mathcal{L}_{m(x)} \neq \emptyset$). Visited cells are assigned a reward of -1 . In addition to the given rewards for target servicing, we introduce rewards for exploration denoted by $r_\varepsilon > 0$ associated with observed cells that do not contain targets, $\mathcal{L}_{m(x)} = \emptyset$. Lastly, all unobserved cells ($m(x) = \ell_{\boxtimes}$) are associated with a reward dependent on the estimated number of targets that can be serviced and the number of unobserved cells. Formally, the designed path-dependent reward is

$$\mathcal{R}(x) = \begin{cases} \mathbb{E}_{\mathcal{M}_x}[r(\mathcal{L}) \mid m(x)] & \text{if } m(x) \neq \ell_{\boxtimes}, \mathcal{L}_{m(x)} \neq \emptyset, \\ & x \notin Vis(path) \\ -1 & \text{if } m(x) \neq \ell_{\boxtimes}, x \in Vis(path) \\ r_\varepsilon & \text{if } m(x) \neq \ell_{\boxtimes}, \mathcal{L}_{m(x)} = \emptyset \\ & x \notin Vis(path) \\ \frac{N \cdot \sum_{\ell \in \mathcal{L}} \mathbf{r}(\ell)}{|\mathcal{X} \setminus Obs| \cdot |\mathcal{L}|} & \text{if } m(x) = \ell_{\boxtimes} \end{cases}$$

where $path$ is the robot's path at the current step (see Alg. 1), and $Obs = \{x \mid m(x) \neq \ell_{\boxtimes}\}$ is the set of observed cells.

2) **Reward Collection:** As the robot moves through the environment, it collects the reward for the current cell and some partial rewards for observing the incremental symbols of the cells within the sensing range. The partial rewards ensure that the robot progresses towards cells with targets.

For each cell x along a path \mathbf{x} to be evaluated, the rewards are collected for all cells $x' \in \mathcal{N}_x^{\leq D}$, i.e., for $\lambda \in (0, 1)$

$$R_x(x') = \begin{cases} \lambda^{-d} \cdot \mathcal{R}(x') & d = \|x' - x\|_1 \leq D \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

In case a cell x' is observed from multiple cells along \mathbf{x} , then the maximum reward is collected. Let $Obs_{\mathbf{x}}(x') = \{x \in Vis(\mathbf{x}) \mid x' \in \mathcal{N}_x^{\leq D}\}$. Formally, if $|Obs_{\mathbf{x}}(x')| > 1$, then the collected reward is $\max_{x \in Obs_{\mathbf{x}}(x')} R_x(x')$.

D. Maximum Reward Path Planning

Problem II.1 can now be cast as finding a path in \mathcal{G}_ε from the source node $(x_0^T, \mathbf{E}_\varepsilon)$ to one of the goal states in F_ε whose projection on \mathcal{T} maximizes the total collected rewards. We obtain the solution via a mixed-integer linear program (MILP) with an objective

$$\max_{y_x} \sum_{x \in X} \left\{ \max_{x' \in \mathcal{N}_x^{\leq D}} R_{x'}(x) \cdot y_{x'} \right\} \quad (5)$$

subject to the constraints described below.

$$\sum_{v \in \mathcal{N}_u^-} \zeta_{u,v} - \sum_{v \in \mathcal{N}_u^+} \zeta_{u,v} = \begin{cases} 0, & u \neq s, u \neq t \\ 1, & u = t \\ -1, & u = s \end{cases}$$

where \mathcal{N}_u^- and \mathcal{N}_u^+ denote the predecessors and successors of node $u \in V_\varepsilon$, respectively. This constraint captures flow conservation as follows: $\zeta_{u,v} \in \mathbb{B}$ is a decision variable indicating whether the edge (u, v) is part of the solution path. s is $(x_0, \mathbf{E}_\varepsilon)$, and t is a virtual state such that all states F_ε are connected to t .

$$z_u = \begin{cases} \sum_{v \in \mathcal{N}_u^+} \zeta_{u,v}, & u \neq t, \\ \sum_{v \in \mathcal{N}_u^-} \zeta_{u,v}, & u = t, \end{cases}, \quad \forall u \in V_\varepsilon$$

$$y_x \leq \sum_{u=(x,e) \in V_\varepsilon} z_u, \quad \forall x \in X$$

$$y_x \geq z_u, \quad \forall u = (x, e) \in V_\varepsilon \forall x \in X$$

where $y_x \in [0, 1]$ and $z_u \in \mathbb{B}$ are decision variables. These constraints together consider whether a node u is visited so as to collect the reward for its projection on \mathcal{T} , i.e., state x . This formulation is a modified version of the standard optimal flow algorithms on DAGs where the modifications take into account the cells within the sensing range.

Discussion. The decision variables of (5) are defined on \mathcal{G}_ε . To ensure efficient execution, the product model is pruned at each step. Finally, the energy budget allocation can be informed by the robot's knowledge at runtime. However, deciding when to reallocate the energy is a non-trivial decision and is a topic for future research.

Feasibility. Alg.1 is recursively feasible by construction. However, no guarantees can be provided about optimality.

V. CASE STUDIES

In this section, we demonstrate the efficacy of the proposed decision-making framework applied to a Mars exploration scenario. We then proceed to evaluate the planning with the EISP algorithm with a baseline case where complete information about the environment is available a priori. Finally, we present data on the runtime performance of the MILP defined in (5).

A. Planning

Consider an autonomous robot in a Martian environment deployed to collect samples of Biomarkers and Fossils. Thus, $\mathcal{L} = \{\text{Fossil}, \text{Biomarker}\}$ Fig. 7 shows the perception refinement. In this case study, the sensing range is $D = 2$ cells. On an 8×8 grid, the robot is tasked to go from $x_{init} = (0, 0)$ to $x_{DOCK} = (7, 6)$ with $\mathbf{E} = 22$, and $\mathbf{e}(\text{Fossil}) = 3$, $\mathbf{e}(\text{Biomarker}) = 2$, and $\mathbf{r}(\text{Fossil}) = 8$, $\mathbf{r}(\text{Biomarker}) = 6$. For testing, we plant targets at $x_{(2,3)}, x_{(4,7)}, x_{(7,2)}$ with $h(x_{(2,3)}) = \text{Fossil}$, $h(x_{(4,7)}) = \text{Fossil}$ and $h(x_{(7,2)}) = \text{Biomarker}$ and these are hidden from the robot. Each transition in the grid consumes 1 unit of energy.

Fig. 5 shows the evolution of the rewards at various instances with respect to the robot's current location, past knowledge, and some limited information about currently visible cells. The figure shows the rewards at the top and

the planned path (dark blue nodes, black arrows) as well as the cells within the sensing range (cyan). The targets are shown using purple diamonds. At $t = t_0$, since no targets can be observed, the rewards are uniformly distributed over the cells outside the robot's sensing range. Subsequently, the robot's tracking model is updated at each time step e.g., at $x_{(2,1)}$, $m(x_{(2,3)}) = \text{Rock}$ and so on. Note that, at $t = t_{11}$, even though the robot acquires partial information about the target at $x_{(4,7)}$, it is unable to re-plan and collect the sample due to low remaining energy. The robot finally reaches the DOCK at t_{12} .

B. Empirical Evaluation

To the authors' best knowledge, there are no existing exploration algorithms with incremental-resolution symbolic perception. Thus, for empirical evaluation of our approach, we resort to a baseline case where full information about the environment is assumed.

We refer to the baseline with full information as *F.I.* and our model with no initial information as *N.I.I.*. To set up the testing scenarios, the number of events of each type $\{\nu_i\}_{i \in \mathcal{L}}$ are sampled at random from $\mathcal{D}(\cdot)$, where $\mathcal{D}(\cdot)$ is chosen to be a geometric distribution and the total number of events is $\nu = \sum_{i \in \mathcal{L}} \nu_i$. The event locations are generated randomly using *Shuffle* method given the grid size where first ν locations are chosen. For each test case, the event locations and the number of events of each type are same for *F.I.* and *N.I.I.*. The perception refinement, target symbols, energy values \mathbf{e} and target rewards \mathbf{r} are same as the previous case study. We vary the grid sizes, energy budget, total number of targets present as well as event locations and evaluate the empirical mean regret of not having the full information. For each grid size and ν , we generate 10 scenarios corresponding to different locations of events.

Evaluation Criteria. We use regret to quantify the effectiveness of the proposed approach. The expected regret is calculated in terms of total expected reward for collecting samples¹ and is given by

$$\mathbb{E}[\text{Regret}] = \mathbb{E}_{\nu, \ell' \in \mathcal{L}} [\mathcal{R}_{serv}(\ell')_{F.I.}] - \mathbb{E}_{\nu, \ell'' \in \mathcal{L}} [\mathcal{R}_{serv}(\ell'')_{N.I.I.}].$$

Table I summarizes the evaluations across multiple combinations of the varying entities. Despite having the full information about the event locations, *F.I.* may not collect all targets due to the energy constraints.

TABLE I: Regret Evaluation

Case No.	Grid Size	\mathbf{E}	No. of targets present	Targets serviced		Mean Regret
				F.I.	N.I.I.	
1	4×4	15	3	2.8	1.5	9.6
2	5×5	17	3	2.2	1.5	5.4
3	5×5	20	4	2.8	2.4	3.2
4	6×6	23	4	3.4	2.4	7.8
5	8×8	29	6	4.4	3.8	4
6	8×8	18	1	1	0.6	3.2
7	8×8	20	2	1.7	1.3	2.8

¹Note that, even though *N.I.I.* considers incremental rewards for observing incremental symbols of targets, this case study only considers rewards for targets collected to ensure objective comparison.

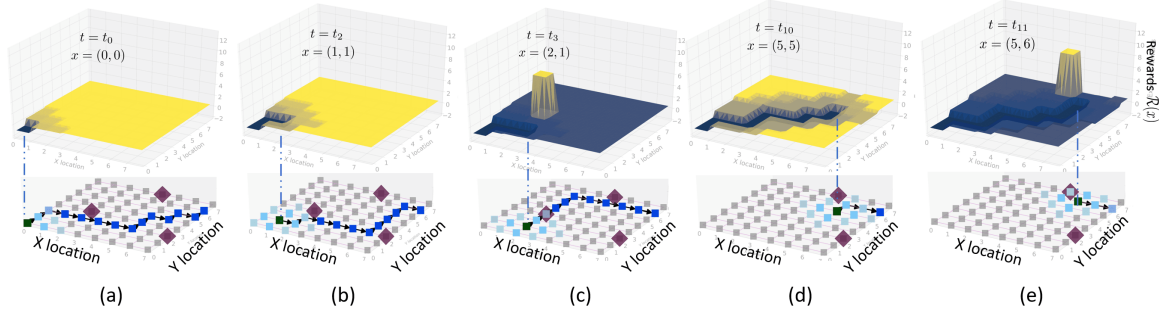


Fig. 5: The figure shows the rewards over all grid cells given current robot location x and the planned path to goal (blue markers) at various time instances between t_0 and t_{11} . The cells within the sensing range of the robot are shown in cyan and the targets are shown in purple diamonds. These figures depict the evolution in rewards during mission execution.

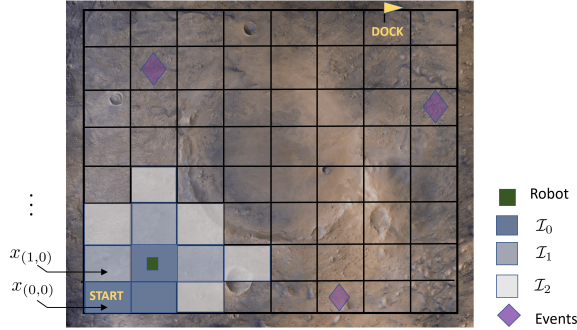


Fig. 6: \mathcal{T} for the Mars Exploration Problem. The robot is tasked to go from START to DOCK while locating as many targets as possible. The cells within the sensing range of the robot are color-coded w.r.t. the refinement MDP shown in Fig. 7. e.g., at the current state $x_{(1,1)}$, the robot exactly knows the label “No Sample”.

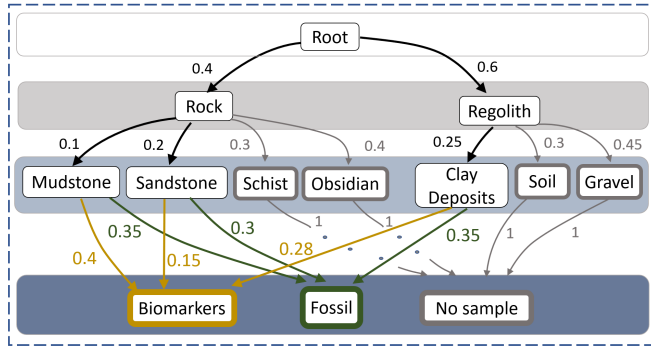


Fig. 7: A refinement for locating the targets of interest viz, biomarkers and fossil samples. The black arrows show transition probabilities to incremental symbols and the yellow, green arrows as well as the corresponding colored probability values indicate the probability of finding a biomarker and fossil, respectively. Finally, the gray arrows and states lead to event of not finding any sample and thus, are eliminated for clarity of presentation.

C. Runtime Performance

In this section, we evaluate the runtime performance of solving the objective function defined in Eq. 5 using an off-the-shelf optimization tool Gurobi [28]. These case studies were performed on Dell Precision 3640 Intel i9-10900K with 64 GB RAM using python 3.9.7. We vary the grid sizes as well as the energy budget to compare the number of binary and continuous variables created in the formulation and

report the time taken by Gurobi to find an optimal solution for the first iteration of the MILP problem as shown in Table II. The product size refers to the number of transitions in the product graph. In order to encode the constraints presented in Eq. 5, a binary variable is defined for each transition in the product graph. Furthermore, linearizing the objective function requires an auxiliary binary variable to be defined for each node in the product such that its projection on \mathcal{T} is within the sensing range D of the current cell. This introduces a large number of decision variables. These values are reported at the beginning of model creation which after pre-solving, reduce drastically.

TABLE II: Runtime Performance

Grid Size	E_{ε}	$ \Xi_{\varepsilon} $	Continuous	Binary	Time (s)
4×4	11	108	7242	7336	0.185
5×5	13	220	23686	23883	0.539
6×6	16	450	72668	73085	1.707
7×7	18	714	159600	160268	3.700
8×8	20	1064	314795	315798	6.965
9×9	21	1512	572770	574204	12.360

VI. CONCLUSION

This work presents a decision-making framework for energy-constrained autonomous exploration with incremental-resolution symbolic perception without any knowledge of targets. We define the abstraction models for encapsulating robot motion, perception, and observation history as the robot explores the environment. Our method casts the problem as an instance of reward maximization problem implicitly integrating the energy constraints within the models. Updating the rewards over the environment at each step, a modified optimal flow problem is solved using MILP. The empirical results obtained via case studies demonstrate the efficacy of the proposed planning framework.

REFERENCES

- [1] S. Wakayama and N. Ahmed, “Active inference for autonomous decision-making with contextual multi-armed bandits,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*.
- [2] M. T. Spaan, “Cooperative active perception using pomdps,” in *AAAI 2008 workshop on advancements in POMDP solvers*, 2008.
- [3] H. Ma and J. Fu, “Attention-based probabilistic planning with active perception,” in *Intl. Conf. on Robotics and Automation*, pp. 8–14, 2021.

- [4] K. Otsu, A.-A. Agha-Mohammadi, and M. Paton, "Where to look? predictive perception with applications to planetary exploration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 635–642, 2017.
- [5] G. Liu, D. Kamale, C.-I. Vasile, and N. Motee, "Symbolic perception risk in autonomous driving," *American Control Conference*, 2023.
- [6] D. Kamale, S. Haesaert, and C.-I. Vasile, "Cautious planning with incremental symbolic perception: Designing verified reactive driving maneuvers," *IEEE Intl. Conf. on Robotics and Automation*, 2023.
- [7] K. Zheng, Y. Sung, G. Konidaris, and S. Tellex, "Multi-resolution pomdp planning for multi-object search in 3d," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 2022–2029, 2021.
- [8] K. Seiler, H. Kurniawati, and S. Singh, "An online and approximate solver for pomdps with continuous action space," in *IEEE Intl. Conf. on Robotics and Automation*, pp. 2290–2297, 2015.
- [9] S. Haesaert, R. Thakker, R. Nilsson, A. Agha-Mohammadi, and R. M. Murray, "Temporal logic planning in uncertain environments with probabilistic roadmaps and belief spaces," in *IEEE Conference on Decision and Control*, pp. 6282–6287, 2019.
- [10] P. Nilsson, S. Haesaert, R. Thakker, K. Otsu, C.-I. Vasile, A.-A. Agha-Mohammadi, R. M. Murray, and A. D. Ames, "Toward specification-guided active mars exploration for cooperative robot teams," in *Robotics: Science and Systems*, 2018.
- [11] C.-I. Vasile, K. Leahy, E. Cristofalo, A. Jones, M. Schwager, and C. Belta, "Control in belief space with temporal logic specifications," in *2016 IEEE 55th Conference on Decision and Control (CDC)*.
- [12] K. Leahy, E. Cristofalo, C.-I. Vasile, A. Jones, E. Montijano, M. Schwager, and C. Belta, "Control in belief space with temporal logic specifications using vision-based localization," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 702–722, 2019.
- [13] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon" next-best-view" planner for 3d exploration," in *IEEE Intl. Conf. on Robotics and Automation*, pp. 1462–1468, 2016.
- [14] J. L. S. Rincon and S. Carpin, "Time-constrained exploration using toposemantic spatial models: A reproducible approach to measurable robotics," *IEEE Robotics & Automation Magazine*.
- [15] M. Cai and C.-I. Vasile, "Safe-critical modular deep reinforcement learning with temporal logic through gaussian processes and control barrier functions," *arXiv preprint arXiv:2109.02791*, 2021.
- [16] M. Cai, E. Aasi, C. Belta, and C.-I. Vasile, "Overcoming exploration: Deep reinforcement learning for continuous control in cluttered environments from temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2158–2165, 2023.
- [17] B. Lindqvist, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, "Exploration-rrt: A multi-objective path planning and exploration framework for unknown and unstructured environments," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 3429–3435, 2021.
- [18] C. I. Vasile, X. Li, and C. Belta, "Reactive sampling-based path planning with temporal logic specifications," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 1002–1028, 2020.
- [19] D. Kamale, E. Karyofylli, and C.-I. Vasile, "Automata-based optimal planning with relaxed specifications," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 6525–6530, 2021.
- [20] X. Ding, M. Lazar, and C. Belta, "Ltl receding horizon control for finite deterministic systems," *Automatica*, vol. 50, no. 2, 2014.
- [21] D. Aksaray, C.-I. Vasile, and C. Belta, "Dynamic routing of energy-aware vehicles with temporal logic constraints," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*.
- [22] L. Carlone and D. Lyons, "Uncertainty-constrained robot exploration: A mixed-integer linear programming approach," in *IEEE International Conference on Robotics and Automation*, pp. 1140–1147, 2014.
- [23] K. Leahy, Z. Serlin, C.-I. Vasile, A. Schoer, A. M. Jones, R. Tron, and C. Belta, "Scalable and robust algorithms for task-based coordination from high-level specifications (scratches)," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2516–2535, 2021.
- [24] S. Saha and A. A. Julius, "An milp approach for real-time optimal controller synthesis with metric temporal logic specifications," in *2016 American Control Conference (ACC)*, pp. 1105–1110, IEEE, 2016.
- [25] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. A. Castellanos, "Active policy learning for robot planning and exploration under uncertainty," in *Robotics: Science and systems*, pp. 321–328, 2007.
- [26] Z. Qian, J. Fu, and J. Xiao, "Autonomous search of semantic objects in unknown environments," *arXiv preprint arXiv:2302.13236*, 2023.
- [27] O. Peltzer, A. Bouman, S.-K. Kim, R. Senanayake, J. Ott, H. Delecki, M. Sobue, M. J. Kochenderfer, M. Schwager, J. Burdick, and A.-a. Agha-mohammadi, "Fig-op: Exploring large-scale unknown environments on a fixed time budget," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 8754–8761, 2022.
- [28] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020.