Energy Efficient Planning for Repetitive Heterogeneous Tasks in Precision Agriculture

Shuangyu Xie and Dezhen Song

Abstract— Robotic weed removal in precision agriculture introduces a repetitive heterogeneous task planning (RHTP) challenge for a mobile manipulator. RHTP has two unique characteristics: 1) an observe-first-and-manipulate-later (OFML) temporal constraint that forces the order of the two different tasks for each target and 2) energy savings from efficient task collocation to minimize unnecessary movements. Stochastically, RHTP is a renewal process by nature. According to the Renewal Reward Theorem, the expected energy usage per task cycle is the long-run average. Traditional task and motion planning focuses on feasibility rather than optimality due to the unknown object and obstacle position prior to execution. However, the known target/obstacle distribution in precision agriculture allows minimizing the expected energy usage. We propose a task space partition-based approach by computing all possibilities of task multiplexing and its probabilities while considering the robot reachability constraint. We have implemented and tested our RHTP algorithm compared to its counterpart in simulation based on real field data. The results show a significant improvement in path length, number of robot stops, overall energy usage, and number of replans.

I. INTRODUCTION

Deploying robotic systems in precision agriculture reduces labor requirements and avoids over-application of chemicals. We present a robotic weed removal planning algorithm to guide the mobile manipulator (Fig. 1) to traverse the field and precisely burn the key growth spot of each weed using a flaming torch, which is environmentally friendly and gains popularity in applications [1], [2]. Such tasks are often repetitive in nature, and energy usage can be reduced from task collocation if planned efficiently. Named repetitive heterogeneous task planning (RHTP) problem, the weed removal process is a renewal-reward process. According to the Renewal Reward Theorem [3], the expected energy usage per task cycle is the long-run average. Consequently, it is imperative to minimize the expected energy usage per cycle when determining the robot's action sequence and paths. Traditional integrated Task and Motion Planning (TAMP) [4] typically addresses single-instance cases, prioritizing solution feasibility over optimality due to uncertain object and obstacle positions. However, precision agriculture often provides known spatial distributions of targets, enabling optimization of expected energy usage for repetitive tasks.

Robotic weed flaming applications also pose new challenges. Targeting peripheral leaves or branches may not effectively suppress or eliminate weeds [1]. Since real-time

S. Xie, and D. Song are with Department of Computer Science and Engineering, Texas A&M University. D. Song is also with Department of Robotics, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI) in Abu Dhabi, UAE. Corresponding author: Dezhen Song. Email: `dezhen.song@mbzuai.ac.ae`.
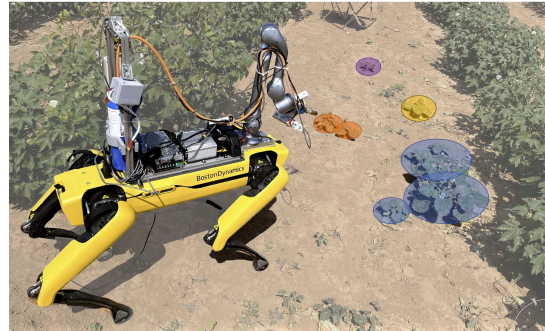

Fig. 1: A weed removal robot operating in a cotton field, where the target weeds are highlighted as colored regions. Weed regions sharing the same color belong to the same cluster.

high-resolution field maps are often inaccessible, robots must conduct close-range inspections to precisely locate each weed's growth point for effective flaming. This approach introduces two key task constraints: i) each task requires the robot to first observe the target, revealing its precise position before executing the manipulation step, and ii) multiple clustered targets reduce robot base movements to save energy and increase operational speed if planned carefully. These factors result in heterogeneous tasks with specific sequencing requirements and opportunities for optimized movement. Similar properties are common in various precision agriculture applications, including harvesting and spraying.

We solve the RHTP problem by minimizing the expected energy usage per cycle. We propose a partition-based task space approach by computing all possibilities/probabilities of task multiplexing while considering the robot's reachability constraint. This transforms the original problem into a region-based optimization problem that focuses on spatial task multiplexing and sequencing. We implemented and tested our proposed algorithm in simulations using data collected from a cotton field with our custom platform (Fig. 1). The results show that our RHTP algorithm significantly reduces the path length, the number of robot stops, the energy cost, and the number of replans compared to its counterpart. Notably, these improvements become more substantial as the target density increases, which is desirable for practical agricultural scenarios.

II. RELATED WORK

We view our RHTP problem as a special case of the long-horizon planning (LHP) problem [9]. LHP is a challenging planning problem that generates robot behaviors or trajectories to achieve complex tasks or navigation objectives in an environment of uncertainty [10]. Our LHP problem is related

TABLE I: LHP & complex planning problem comparison. * means approximation to the optimal.

Problem & Algorithm	Task	Platform	Scene Knowledge	Metric & Optimality
Sensor Planning [5]	N	UAV+UGV	Complete	Optimal trajectory*
Tabletop Object Rearrangement [6]	M	Mobile Manipulator	Complete	Optimal p&p sequence
Blindfolded Traveler Problem [7]	PO + N	Mobile robot	Partial, Incremental	Optimal trajectory*
TAMP w. Partial Observation [8]	PO+N+M	Mobile Manipulator	Partial, Incremental	Feasible task plan, optimal state trajectory*
RHTP (ours)	AP+N+M	Mobile Manipulator	Partial, Incremental	Long-run optimal task&state*

to informative path planning [5], multi-object manipulation [11], [12], navigation with uncertainty [7], [13], and the TAMP framework [14], [4], [15]. We compare our work with closely related works and summarize the comparison in Table I as detailed below.

Tasks and Robot Platform: For LHP, the task can be classified as observation (O), manipulation (M), and navigation (N). Depending on whether the observation task is required to actively acquire information, it can be divided into passive observation (PO) and active perception (AP) [16], [17]. The types of robot platforms and the consequent task capabilities depend on each other. For tasks that require both navigation and manipulation ability, mobile manipulators have been widely deployed due to their high versatility [18], [19]. With the combination of active perception and manipulation, the hand-eye system [20] enables the robot to switch between near-range observation and the required manipulation tasks.

Scene Knowledge: Significant advancements have been made in planning algorithms for scenarios with complete prior scene knowledge, even for tasks involving complex motion primitives [21] or long-action sequencing in multiple-object manipulation [6] (Table I, row 2). Non-adaptive informative planning, such as sensor planning in [5], assumes deterministic action outcomes and complete scene information for planning (Table I, row 1). When prior scene knowledge is incomplete, planning must incorporate perception alongside navigation or manipulation, as the robot needs to collect information incrementally from the scene during the execution of the trajectory [8], [22]. For partially observed environments with occlusions, Garrett et al. [8] developed an online replanning algorithm based on TAMP. Their method employs a particle filter to represent object pose beliefs, reducing belief states to deterministic values through self-loop action determination (Table I, row 4). Contact-based planning in blindfolded traveler's problem [7] updates the beliefs of obstacles while planning the best expected route (Table I, row 3).

Metrics and Optimality: Planning performance is often evaluated based on metrics such as time to complete a task, task success rate, trajectory length, etc. Depending on the decision variable, they can be classified as two types: i) task-level metrics refer to metrics to assess task assignment or action sequence as decision variables. For a given task plan, if changing symbolic action, adding / removing action, or changing the order of actions does not improve overall performance, the task plan is optimal [23]. ii) motion/state-level optimality is defined on robot motion, trajectory, and poses. For navigation-type tasks, the focus

of the algorithms is to find the shortest travel distance using approximation algorithms [5], [7]. ORLA* [6] for the tabletop rearrangement problem find the sequence of pick-and-place (p&p) actions that minimizes execution time. TAMP often solves optimality at the continuous motion/state level [24], but does not optimize the efficiency of the plan at the discrete task/action level [4]. Although [8], [22] share observation and manipulation steps, their algorithms focus on the feasibility of completing a complex multistage task in a single scene instance. In contrast, our problem is focused on minimizing energy usage in long run average due to our application characteristics. Additionally, our focus is on the aspect of spatial task multiplexing rather than motion/state-level planning, as the latter is often straightforward in an open agricultural field.

RHTP extends the single target coupled active perception and manipulation problem (CAPM) introduced in [25]. While CAPM introduces the Observe-First-Manipulate-Later (OFML) constraint for single targets, RHTP advances this concept by addressing multiple clustered targets and incorporating spatial task multiplexing. This extension significantly broadens the applicability and efficiency of the approach in complex, multitarget environments.

III. BACKGROUND AND PROBLEM FORMULATION

Regenerative point: As the robot repetitively performs weed removal tasks in the field, it periodically captures overhead images from the row center line by elevating its camera to provide an overhead view. We refer to weeds as targets. Due to the perspective limit, these images provide only a low-resolution scene description, containing the target regions of interest (TROI) depicted as colored circular regions in Fig. 1. TROIs are typically identified through object detection algorithms applied to these overhead images. The planning algorithm needs to find an energy-efficient plan to process all known TROIs. The moment when an overhead photo is taken serves as the natural regenerative point in this renewal process. We formulate an instance of the RHTP problem.

A. Assumptions

The main assumptions are as follows.

a.1 The mobile platform is holonomic.

a.2 No target is surrounded by obstacles and becomes unreachable.

a.3 The robot does not move its base and arm simultaneously to ensure stability.

a.4 The arm is in a rest position while the base is moving.

a.5 The energy usage of arm movement for each observation / flaming manipulation task can be approximated as constants.

Assumptions a.1 and a.2 are common in early stage weed removal applications where limited weed and crop heights do not impede robot mobility. In addition, quadruped robots ensure holonomic motion. These assumptions are made to reduce the complexity of the problem and can be relaxed in the future. Assumptions a.3 and a.4 enhance stability and simplify obstacle avoidance. Assumption a.5 is valid for repetitive tasks.

B. Scene Description

Denote the robot workspace $\mathcal{W} \subset \mathbb{R}^3$ with the x-y plane overlap with the horizontal ground plane and the z axis pointing upward. Given n targets with index set $\mathcal{I} = \{1, ..., n\}$, TROI and manipulation point of interest (MPOI) for the target $i \in \mathcal{I}$ are defined as follows.

T_i Target i's TROI defined by the geometric center at $X_i^c = [x_i, y_i, 0] \in \mathcal{W}$ of the target and its radius r_i. $T_i := \{X = [x, y, z] : \|X - X_i^c\|_2^2 \leq r_i^2, z \geq 0\} \subset \mathcal{W}$.

X_i Target i's MPOI, $X_i \in T_i$. Note that TROI's geometric center X_i^c does not necessarily overlap with X_i.

For each target i, a reasonable model of the distribution is a uniform distribution over T_i since X_i could be anywhere in T_i. However, the weed growth point is likely to be located in the center of T_i due to biological symmetry. We model X_i as a truncated normal distribution: $X_i \sim \mathcal{N}_T(X_i^c, \Sigma_i, r_i)$, $1 \leq i \leq n$ where X_i^c is the center of T_i, Σ_i is the covariance matrix, and r_i is the truncation radius. The initial belief space of the n MPOIs is thus defined as:

Definition 1. *(MPOI Belief Space) For a vector of n MPOIs, $\mathbf{X} = [X_1, ..., X_n]$, its belief space $\mathcal{B}_\mathbf{X}$ can be represented as $\mathbf{X} \sim \mathcal{B}_\mathbf{X}$ with*

$$\mathcal{B}_\mathbf{X} = [\mathcal{N}_T(X_1^c, \Sigma_1), ..., \mathcal{N}_T(X_n^c, \Sigma_n)]. \qquad (1)$$

Each close-up observation collapses the corresponding i-th belief $\mathcal{N}_T(X_i^c, \Sigma_i, r_i)$ into deterministic values.

C. Problem Formulation

We formulate the RHTP with mobile manipulators for a set of weed-flaming tasks as the base placement sequence selection and the task assignment problem. Define $\mathcal{K} = \{0, ..., K\}$ as the time sequence index set. $\mathbf{x}_{b,k} \in \mathcal{X}_b$ denotes base pose at the k-th stop. The task plan can be represented as the sequence of base stops and the target index set assigned to the k-th base stop: $\pi = \{(\mathbf{x}_{b,k}, \mathcal{I}_k)\}_{0:K}$, where $\mathcal{I}_k \subset \mathcal{I}$.

1) Observation/Manipulation Task Constraints: For our mobile manipulator with hand-eye system, the observation reachability constraint forces the arm to approach the target i so that T_i is fully covered in the camera image with sufficient resolution to identify X_i. The manipulation reachability constraint determines whether the manipulator can reach the target while avoiding self-collision or an inability to perform the weed-flaming task without damaging the robot.

Given a known manipulator configuration, we model the observation/manipulation reachability constraint as a binary function with $\mathbb{1}_{O/M}(\mathbf{x}_{b,k}, T_i, \bar{X}_i) = 1$ which means that the constraint is satisfied. The detailed formulation will be explained later.

Another type of constraint is observe-first-and-manipulate-later (OFML) temporal constraint that forces the order of the two different tasks:

Definition 2. *(OFML) For the target i, let $k_\tau, k_\iota \in \mathcal{K}$ be time indices for its observation and manipulation actions in π, respectively. OFML can be described as $k_\tau \leq k_\iota, \forall i \in \mathcal{I}$.*

2) Energy cost: We evaluated plans π based on energy consumption due to limitations of battery capacity onboard. Our goal is to compare the energy costs of different task sequences instead of computing the precise execution costs. Given Assumption a.5, the arm energy does not vary with the task sequence. Therefore, we focus on the energy consumption from the robot base movement. We follow the energy modeling in [26] using a combination of a fixed initial starting cost for the base and a variable energy cost, but other energy cost models can also be used. Denote $e_d(\mathbf{x}_{b,k-1}, \mathbf{x}_{b,k})$ as the obstacle-free travel distance between $\mathbf{x}_{b,k-1}$ and $\mathbf{x}_{b,k}$, for a plan sequence, the normalized cost is

$$c(\pi) = K + \gamma \sum_{k=1}^{K} e_d(\mathbf{x}_{b,k-1}, \mathbf{x}_{b,k}). \qquad (2)$$

where γ is the energy cost coefficient, a ratio between the starting cost and the variable costs of the base, $\gamma > 0$.

Problem 1 (RHTP). *Given the start state $\mathbf{x}_s$, goal state $\mathbf{x}_g$, and a set of TROIs $\{T_i : i \in \mathcal{I}\}$, sequentially find a task plan π and for action and key states to obtain MPOI set $\{X_i : i \in \mathcal{I}\}$ to guide and execute the subsequent manipulation task with the minimum expected energy cost:*

$$\min_{\mathbf{a}, \mathbf{x}} \quad \mathbb{E}_{\mathbf{X} \sim \mathcal{B}_\mathbf{X}}[c(\pi)] \qquad (3)$$

$$s.t. \quad \bigwedge_i \mathbb{1}_{O/M}(\mathbf{x}_{b,k}, T_i, \bar{X}_i) = 1, \forall i \in \mathcal{I}, \qquad (4)$$

$$\mathbf{x}_{b,0} = \mathbf{x}_s, \mathbf{x}_{b,k_{max}} = \mathbf{x}_g, \text{ and OFML in Def. 2} \qquad (5)$$

IV. ALGORITHM

An effective way to reduce energy is to group the targets using clustering. Two targets are considered to be in the same cluster if there exist mobile manipulator configurations that can observe/manipulate either target without moving the base. As an example, Fig. 1 colors the clusters differently. Clustering enables spatial task multiplexing (STM) of robot base motions. It is clear that STM can reduce base movements, reduce energy usage, and increase speed.

Meanwhile, the OFML constraint in Def. 2 prevents deterministic planning before MPOI is observed. We need to employ the MPOI belief space in Def. 1 to construct the task sequence. For each cluster, a wise strategy is to increase the likelihood of spatial multiplexing by choosing the optimal base positions. As a result, there is a combinatorial nature

to this problem in the iterative process. We begin with introducing the probabilistic target reachability map (PTRM).

A. Probabilistic Target Reachability Map

The construction of the PTRM involves converting the MPOI belief $\mathcal{B}_{\mathbf{X}}$ into the probability that the mobile manipulator can reach a target from a given base pose $\mathbf{x}_b \in \mathcal{X}_b$, as defined in (4). This constraint is inherently probabilistic due to the random distribution of unobserved targets.

The robot base state constraints can be constructed through the circular approximation [27] by inverse kinematics of arm configurations. The reachable space for the robot base $\mathcal{R}_o(T_i) \subset \mathcal{X}_b$ to observe T_i can be approximated to a compact, closed donut-shaped region due to camera coverage and resolution constraints. Similarly, given X_i, the reachable space $\mathcal{R}_m(X_i) \subset \mathcal{X}_b$ for the robot base enabling manipulation is also a closed donut-shaped region due to self-collision and kinematic constraints. Detailed derivations can be found in [25]. Eq. (4) is satisfied if $\mathbf{x}_b \in \mathcal{R}_o(T_i)$ and $\mathbf{x}_b \in \mathcal{R}_m(X_i)$ when observing X_i. In general cases, the feasible manipulation region $\mathcal{R}_m(X_i)$ is smaller than the feasible observation region $\mathcal{R}_o(T_i)$ because the camera on board can often cover a much larger region than the arm can reach; as analyzed in [25].

Considering that the probability on the estimated $\hat{X}_i$ is truncated and is limited in a small region of TROI, we further approximate the reachability constraints such that all the base states that satisfy the manipulation constraint also satisfy the observation constraint:

$$\mathcal{R}_m(\hat{X}_i) \subset \mathcal{R}_o(T_i), \forall \hat{X}_i \in T_i. \tag{6}$$

This approximation allows us to formulate $\mathcal{R}_m(\hat{X}_i)$ as the mutual base region for the joint observation/manipulation (O/M) task. This leads to the following new formulation of reachability constraint on the base state:

$$\mathbb{1}_{\text{O/M}}^b(\mathbf{x}_b, \hat{X}_i) = \begin{cases} 1, & \text{if } \mathbf{x}_b \in \mathcal{R}_m(\hat{X}_i) \\ 0, & \text{Otherwise.} \end{cases} \tag{7}$$

Under the simplification, we replace reachability constraint (4) with (7) for base placement in our RHTP problem.

Due to the OFML constraint, the reachable region for the manipulation task is nondeterministic. We calculate the probability of performing a successful task in pose $\mathbf{x}_b \in \mathcal{X}_b$, based on the current belief space $\mathcal{B}_{\mathbf{X}}$. Next, we are ready to introduce the initialization and update of the PTRM.

Initialization: at the starting point, the probability of base position $\mathbf{x}_b$ that allow the robot to reach the target i to execute a successful task is,

$$p_i(\mathbf{x}_b) := p(\mathbf{x}_b \in \mathcal{R}_m(\hat{X}_i)|\hat{X}_i) \tag{8}$$
$$= \int f_T(\hat{X}_i; X_i^c, \Sigma_i, r_i) \mathbb{1}_{\text{O/M}}^b(\mathbf{x}_b, \hat{X}_i) d\hat{X}_i, \tag{9}$$

where $f_T(\cdot)$ is the probability density function of truncated normal distribution. Since $p(\hat{X}_i)$ is a truncated probability, we define a base work space S_i for target i as base configuration set with non-zero probability to reach the target i as
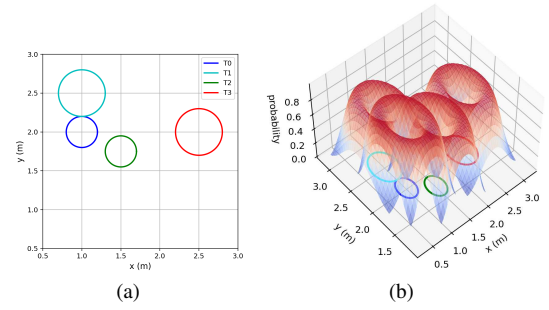

Fig. 2: (a) A running example with 4 targets and the TROIs. (b) PTRM (2D at x-y plane). These probabilities are superimposed on one coordinate system.

$S_i = \{\mathbf{x}_b : p_i(\mathbf{x}_b) > 0\}$, which covers all potential robot base configurations with non-zero probability to complete the task.

Update: During run time, once the target i is perceived and its MPOI is detected after robot movement, X_i becomes deterministic, $p_i(\mathbf{x}_b) := 1, \forall \mathbf{x}_b \in \mathcal{R}_m(X_i)$. Consequently, the MPOI belief $\mathcal{B}_{\mathbf{X}}$ collapses to the observed deterministic value. We also update the base task space $S_i = \mathcal{R}_m(X_i)$.

B. Task Space Partition

The PTRM provides the sets of feasible base states to execute the O/M task for each target. The overlapping regions are the base pose set for the robot where it is possible for the robot to achieve STM. For example, the base state within the region $S_1 \cap S_2$ means that it is possible to multiplex the tasks for both targets if the robot is located in the region. To further investigate the possibilities of STM, we partition the base task space $\mathcal{S}$ into regions based on the task space intersection so that each region corresponds to a unique combination of STM (see Fig. 3).

Definition 3. *(Task Space Partition) A task space partition $\mathcal{Q} = \{Q_j : j \in \mathcal{I}_Q\}$ is constructed from a collection of the base task space sets $\mathcal{S} = \{S_i : i \in \mathcal{I}\}$ such that*

(a) *$\bigcup_{j \in \mathcal{I}_Q} Q_j = \bigcup_{i \in \mathcal{I}} S_i$,*

(b) *If $j \neq l$ then $Q_j \cap Q_l = \emptyset$, and*

(c) *If $Q_j \subseteq S_i$, we label S_i as the parent set of Q_j. For each partition Q_j, it may have multiple parents. Its parent index set is denoted as $\mathcal{I}_j \subset \mathcal{I}$. For two partitions Q_j and Q_l, if $j \neq l$, then $\mathcal{I}_j \neq \mathcal{I}_l$.*

Conditions (a) and (b) in Def. 3 ensure that $\mathcal{Q}$ is a partition. Condition (c) states that each partition Q_j corresponds to a unique STM combination. An example partition is shown in Fig. 3. For each partition of the task space Q_j, we compute the likelihood that the reachability constraint of the task i' is satisfied when the robot base is placed there. Therefore, we define the probability matrix to capture each partition's task success probabilities $\mathbf{P}$ with each element $\mathbf{P}_{i,j}$ calculating the probability of target i can be successfully reached from any base state from the partition region Q_j, i.e. $\mathbf{x}_b \in Q_j$:

$$\mathbf{P}_{i,j} := \frac{\int_{\mathbf{x}_b \in Q_j} p_i(\mathbf{x}_b) d\mathbf{x}_b}{\int_{\mathbf{x}_b \in Q_j} 1 d\mathbf{x}_b} \tag{10}$$
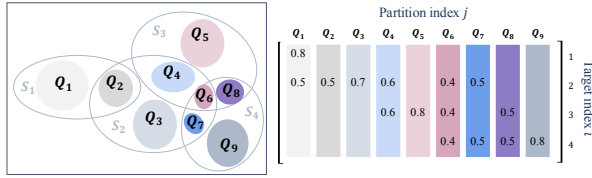
Fig. 3: PTRM partition. Left: a topological visualization of task space partition. Right: partition's task success probabilities matrix. These probabilities are based on location and not normalized against either i or j.

With each $\mathbf{P}_{i,j}$ computed for all partition-target combinations, we obtain the matrix $\mathbf{P}$ to represent the probability of each STM combination. $\mathbf{P}$ is updated each time the PTRM update occurs. The right side of Fig. 3 shows a partition's task success probabilities matrix example.

C. Base Placement Planning

Recall the objective function in (3), we want to minimize the energy $\mathbb{E}[c(\pi)]$. Obviously, we can condition the expected energy on the selected task space partitions which can be viewed as high-level decision variables that direct the robot to stop and execute tasks.

We begin with the probability of reaching the target i for a given partition set. Define the task space variable $\phi = [\phi_1, \ldots, \phi_r]$ corresponding to the partition index set $\mathcal{I}_Q = \{1, ..., r\}$, where binary variable $\phi_j = 1$ means that the partition Q_j is selected and 0 means otherwise. Let M_i be the event in which the robot successfully completes the task for the target i. Then we have conditional probability $p(M_i|\phi)$ derived from the joint failure probability $p(M_i|\phi) = 1 - p(\bar{M}_i|\phi) = 1 - \prod_j (1 - \phi_j \mathbf{P}_{i,j})$. Note that ϕ contains regions that the robot base needs to stop and perform tasks and represents a high-level task plan that corresponds to the output π.

Recalling the energy cost function in (2), we need to minimize the number of selected regions, which reduces the fixed starting cost, since the robot stops at each selected partition. Meanwhile, the distance traveled must be minimized. The RHTP can be viewed as a set cover problem with symmetric traveling salesperson problem (TSP) on the selected task space partition set.

Here, we define the distance traveled between two partition regions Q_j and Q_l as the minimum variable energy cost $e_d(Q_j, Q_l) = \min_{\mathbf{x}_j \in Q_j, \mathbf{x}_l \in Q_l} e_d(\mathbf{x}_j, \mathbf{x}_l)$. We reformulate the cost function to incorporate set selection as decision variables, accounting for both the base stop times and the energy cost of traveling between selected regions from the initial to the final states with a weighting coefficient γ, as shown in Eq. (11).

Let us define the vertex set as the union of the region index set $\mathcal{I}_Q$ with a size of r and the start state with index 0 / end state index with index $r+1$ such that $V = \mathcal{I}_Q \cup \{0, r+1\} = \{0, 1, ..., r, r+1\}$. Q_0 and Q_{r+1} are sets that contain only the start and end poses, respectively, which are also source and sink nodes of the TSP tour on the graph if we treat each partition/region as a node. We define the binary path selection variables $\boldsymbol{\xi}$ where $\xi_{j,k} = 1$ represents the region k visited just after the region j and 0 otherwise. Since $e_d(Q_j, Q_k) =$

$e_d(Q_k, Q_j)$, the TSP is symmetric so that $\xi_{j,k}$ and $\xi_{k,j}$ have the same meaning. The task space partition selection problem as a mixed-integer nonlinear programming (MINLP) problem is formulated as follows,

$$\min_{\phi, \xi} \quad \phi^\mathsf{T}\phi + \gamma \sum_{j=0}^{r+1} \sum_{k=j+1}^{r+1} \xi_{j,k} e_d(Q_k, Q_j) \tag{11}$$

$$\text{s.t.} \quad p(M_i|\phi) \geq \delta, \qquad \forall i \in \mathcal{I}, \tag{12}$$

$$\sum_{k=0}^{j-1} \xi_{k,j} + \sum_{k=j+1}^{r+1} \xi_{j,k} = 2\phi_j, \qquad \forall j \in \mathcal{I}_Q, \tag{13}$$

$$\sum_{j \in \mathcal{I}_Q} \xi_{0,j} = 1, \sum_{j \in \mathcal{I}_Q} \xi_{j,r+1} = 1, \tag{14}$$

$$\sum_{j \in S} \left(\sum_{\substack{k \in \mathcal{I}_Q \setminus S \\ k < j}} \xi_{k,j} + \sum_{\substack{k \in \mathcal{I}_Q \setminus S \\ k > j}} \xi_{j,k} \right) \leq 2,$$
$$\forall S \subset \mathcal{I}_Q, |S| \geq 2. \tag{15}$$

By choosing a proper probability threshold δ, the constraint (12) ensures that the partitions of the selected task space have a high probability to complete all tasks. Motivated by the Dantzig-Fulkerson-Johnson formulation of the TSP, the constraints (13), (14), and (15) ensure a complete tour from the start state to the end state that passes through the selected task space partitions. Eq. (13) requires that two paths be connected for each selected region. Eq. (14) ensures that the start and end nodes should be connected to the path. Eq. (15) is the subtour elimination constraint.

A solution to this MINLP problem can be obtained using Branch-and-Bound solvers [28], [29]. The optimization results are the selection of partitions of the task space indexed by non-zero entries in ϕ and a visit sequence between these partitions specified by the binary path selection variables $\boldsymbol{\xi}$. Through the partition selection variable, we know that the number of stop times for the base is $\kappa = \phi^T\phi$. The sequence of partition ($\mathbf{q}$) to be visited can be generated from the path variable $\boldsymbol{\xi}$ as an index list. For the running example, $\mathbf{q} = [0, 2, 8, 10]$ from Fig. 3.

The final step is to generate a task plan π by finding the base state $\mathbf{x}_b$ of each region in $\boldsymbol{\xi}$ that defines the shortest path in the same order of region selection. Solving the optimal solution $\mathbf{x}_b$ is trivial and we skip the details here for brevity.

V. EXPERIMENTS

A. Experiment Setup

We have implemented the proposed RHTP algorithm using Python 3.8.10 on a Ubuntu 22.04 PC machine with the Intel(R) Core™ i7-10700K CPU @ 3.80GHz. We have tested the algorithm based on field data collected from our custom mobile manipulator (see Fig. 1) to validate the performance for the real application.

Task and System Settings: We have mounted a 6 degree-of-freedom (DoF) Unitree Z1™ manipulator on the Boston Dynamics Spot Mini™ quadruped robot for the weed removal task. The Spot Mini™ carries all attached accessories (i.e. an
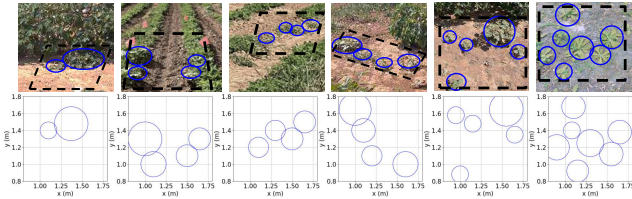
Fig. 4: Sample scene configurations with different target densities from early/late grow stage in cotton field and test bed.

additional computer, flammable fuel container, igniter, fuel pipes, relay controlled valves, control circuit, and a Unitree Z1 arm [2]). We deployed the mobile manipulator in cotton fields to collect data to initialize the simulation used to evaluate our algorithms.

Test Scene Configurations: The test scene is obtained from images collected from the cotton field at different stages of weed growth, as shown in Fig. 4. To validate the performance of the algorithm in different scenarios, we define the target density as the ratio between the number of weeds $|\mathcal{I}|$ and the area of the ground plane of the workspace $\rho = \frac{|\mathcal{I}|}{\text{Area}(\mathcal{W}^g)}$. We have tested a total of 50 scene instances, where for each instance $\text{Area}(\mathcal{W}^g) = 1$ m^2 and the density of the weed varies from 1 to 7.

Baseline Algorithm: The baseline algorithm for comparison is a greedy naive algorithm that extends from the single target CAPM [25] algorithm simply by handling one target at a time until all targets are handled. The target chosen at each step is the closest target that has not been processed. We call this algorithm Naive-CAPM. This can be considered as algorithms based on the hierarchy planning TAMP.

B. Result

1) Speed Test: We have measured the run time of our algorithm, which is 2.2 seconds on average. Since the execution time of actions (base, arm movements, and flaming) is much longer than the planning time, the run time of the algorithm is not the bottleneck in the weed removal application, because the computation can be done while the robot is in motion finishing the previous step.

2) Offline Performance Comparison: We have compared our proposed algorithm (RHTP) with Naive-CAPM for energy usage according to (2) and the number of replans which is the number of times iteration is needed to complete all tasks. For each instance, we sample 1000 possible combinations of MPOIs for the initialization of belief according to (1) where the initial belief parameter is $\Sigma_i = r_i \mathbf{I}_{2\times 2}$ and compute the average result for all metrics. The energy cost and the RHTP parameters are set as $\gamma = 1.12$ and $\delta = 0.7$ based on the hardware testing of our robot, respectively.

First, we compare the two algorithms using all 50 instances in the test data. To separate the effect of density from the uncertainty introduced by prior observation (reflected through the radius), we set the TROI with the same radius.

The plot in Fig. 5 shows the result. When the target density is low $\rho = 1$, the Naive-CAPM and RHTP algorithms perform very similar to each other. This is expected because if there is no cluster, they are essentially the same algorithms.
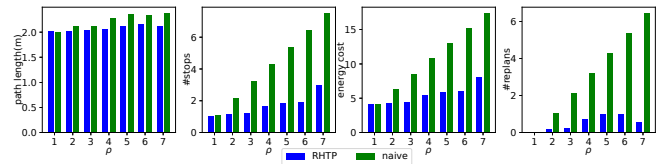

Fig. 5: Performance comparison between the Naive-CAPM and the RHTP algorithms.

As the target density increases, the RHTP has a shorter path length and less number of stops compared to the Naive-CAPM (result trajectory samples are shown in the video attachment). As for the number of replans, the RHTP can achieve less than 1 replan across the board. For the Naive-CAPM method, since one iteration can only find the position for one stop, and the task plan requires the robot to stop for every target, the replan number increases with the density of target increase. In all metrics, the RHTP algorithm outperforms the Naive-CAPM algorithm more significantly as the density increases further. This is very desirable because our RHTP algorithm is more efficient when needed.

To validate the effect of the TROI radius, we choose the instances with the same density $\rho = 5$, where we have the most instance at this density (15 instances). We run both algorithms on this group of instances. We set the radius of the region to $0.15, 0.20, 0.25, 0.30$ meters, respectively. As shown in Fig. 6, both algorithms are affected by the increase in region size because the increase in region size means a greater uncertainty of perception in prescan. In comparison, it is clear that RHTP still outperforms Naive-CAPM in all metrics. The advantage is more significant when the TROI radius (i.e. perception uncertainty) grows larger.
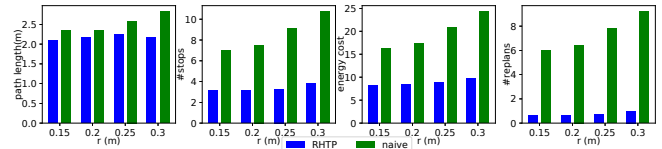

Fig. 6: Performance comparison between the Naive-CAPM and the RHTP at different TROI radius settings. The horizontal axes are TROI radius.

VI. CONCLUSION AND FUTURE WORK

We propose the RHTP algorithm to enable the mobile manipulator to perform repetitive observation and manipulation tasks efficiently in precision agriculture applications. The RHTP algorithm was designed to reduce the long-term average energy consumption rate by minimizing the expected energy use. We reduced the optimization problem to a region-based set-coverage problem by conditioning all possible spatial task multiplexing possibilities in different regions. In experiments, the algorithm demonstrated a significant improvement over its counterpart in all tasks. Moreover, the improvement is more significant when the target density increases, which is desirable. Future work will focus on improving base and arm motion coordination, and developing multi-robot algorithms for repetitive tasks.

REFERENCES

[1] A. Datta and S. Z. Knezevic, "Flaming as an alternative weed control method for conventional and organic agronomic crop production systems: a review," *Advances in agronomy*, vol. 118, pp. 399–428, 2013.

[2] D. Wang, C. Hu, S. Xie, J. Johnson, H. Ji, Y. Jiang, M. Bagavathiannan, and D. Song, "Toward precise robotic weed flaming using a mobile manipulator with a flamethrower," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[3] S. M. Ross, *Introduction to Probability Models*. Academic Press, 13 ed., 2023.

[4] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[5] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.

[6] K. Gao, Y. Ding, S. Zhang, and J. Yu, "ORLA*: Mobile Manipulator-Based Object Rearrangement with Lazy A," *arXiv preprint arXiv:2309.13707*, 2023.

[7] B. Saund, S. Choudhury, S. Srinivasa, and D. Berenson, "The blind-folded robot: A bayesian approach to planning with contact feedback," in *The International Symposium of Robotics Research*, pp. 443–459, Springer, 2019.

[8] C. R. Garrett, C. Paxton, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox, "Online replanning in belief space for partially observable task and motion problems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5678–5684, IEEE, 2020.

[9] D. Driess, J.-S. Ha, R. Tedrake, and M. Toussaint, "Learning Geometric Reasoning and Control for Long-Horizon Tasks from Visual Input," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14298–14305, 2021.

[10] H. Kurniawati, Y. Du, D. Hsu, and W. S. Lee, "Motion planning under uncertainty for robotic tasks with long time horizons," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 308–323, 2011.

[11] K. Gao, S. W. Feng, B. Huang, and J. Yu, "Minimizing running buffers for tabletop object rearrangement: Complexity, fast algorithms, and applications," *The International Journal of Robotics Research*, 2023. to appear.

[12] Z. Pan, A. Zeng, Y. Li, J. Yu, and K. Hauser, "Algorithms and systems for manipulating multiple objects," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 2–20, 2023.

[13] I. S. Mohamed, M. Ali, and L. Liu, "Gp-guided mppi for efficient navigation in complex unknown cluttered environments," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7463–7470, 2023.

[14] J. Wolfe, B. Marthi, and S. Russell, "Combined task and motion planning for mobile manipulation," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 20, pp. 254–257, 2010.

[15] L. P. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1194–1227, 2013.

[16] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[17] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, pp. 177–196, 2018.

[18] S. Thakar, P. Rajendran, A. M. Kabir, and S. K. Gupta, "Manipulator motion planning for part pickup and transport operations from a moving base," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 191–206, 2022.

[19] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," 2022.

[20] I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, "A comparative review of hand-eye calibration techniques for vision guided robots," *IEEE Access*, vol. 9, pp. 113143–113155, 2021.

[21] X. Zhang, Y. Zhu, Y. Ding, Y. Jiang, Y. Zhu, P. Stone, and S. Zhang, "Symbolic state space optimization for long horizon mobile manipulation planning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 866–872, 2023.

[22] J. Muguira-Iturralde, A. Curtis, Y. Du, L. P. Kaelbling, and T. Lozano-Pérez, "Visibility-aware navigation among movable obstacles," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10083–10089, IEEE, 2023.

[23] S.-Y. Lo, S. Zhang, and P. Stone, "Petlon: planning efficiently for task-level-optimal navigation," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 220–228, 2018.

[24] C. Quintero-Peña, Z. Kingston, T. Pan, R. Shome, A. Kyrillidis, and L. E. Kavraki, "Optimal grasps and placements for task and motion planning in clutter," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3707–3713, 2023.

[25] S. Xie, C. Hu, D. Wang, J. Johnson, M. Bagavathiannan, and D. Song, "Coupled active perception and manipulation planning for a mobile manipulator in precision agriculture applications," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[26] L. Hou, L. Zhang, and J. Kim, "Energy modeling and power measurement for mobile robots," *Energies*, vol. 12, no. 1, p. 27, 2018.

[27] T. Sandakalum and M. H. Ang Jr, "Motion planning for mobile manipulators—a systematic review," *Machines*, vol. 10, no. 2, p. 97, 2022.

[28] B. Borchers and J. E. Mitchell, "An improved branch and bound algorithm for mixed integer nonlinear programs," *Computers & Operations Research*, vol. 21, no. 4, pp. 359–367, 1994.

[29] I. Gentilini, F. Margot, and K. Shimada, "The Traveling Salesman Problem with Neighborhoods: MINLP solution," *Optimization Methods and Software*, 2012.