Gathering Data from Risky Situations with Pareto-Optimal Trajectories

Brennan Brodt and Alyssa Pierson

***Abstract*— This paper proposes a formulation for the risk-aware path planning problem which utilizes multi-objective optimization to dynamically plan trajectories that satisfy multiple complex mission specifications. In the setting of persistent monitoring, we develop a method for representing environmental information and risk in a way that allows for local sampling to generate Pareto-dominant solutions over a receding horizon. We propose two algorithms capable of solving these problems: a dense sampling approach and an improved method utilizing noisy gradient descent. Simulation results demonstrate the efficacy of our methods at persistently gathering information while avoiding risk, robust to randomly-generated environments.**

I. Introduction

Risk-aware planning is a flexible formulation of traditional robotic path planning problems that allows for agents to adapt to various hazards within their environments. Risk can model countless challenges autonomous agents encounter while deployed in real world environments, such as: dangerous weather conditions [1], dynamic obstacles [2], and even other malicious agents [3]. With such design flexibility, risk is often used in conjunction with other planners, as a metric to ensure safety during mission operations. Frequently, risk-aware methods are used to plan safe trajectories to target positions [4], however, one under explored application is in the realm of persistent monitoring. Persistent monitoring is the process of continuously controlling agents to complete mission specifications over an arbitrarily long time horizon [5]. This problem has great value to control theory, with applications in particle tracking [6], network monitoring [7], and environmental surveying [8]. This paper proposes a framework in which the persistent monitoring and risk-aware planning problems can be solved simultaneously using techniques from multi-objective optimization.

Our work is motivated by recent results in wildlife monitoring that suggest autonomous agents are more effective in carrying out ecological surveys than traditional methods, but only when proper precautions are taken to prevent adverse responses from the wildlife [9], [10]. With our method, environmental information can signify survey locations, such as herds or nests, while risk can signify obstacles or animals to avoid. Additional models of risk could be included to represent species specific response patterns as, for example, ground mammals typically respond differently to autonomous agents [11], [12] than avians [13]. Further, our method generalizes to a much broader scope of robotics problems, such as in monitoring hazardous environments, inspecting infrastructure in uncertain or dynamic environments, and surveying locations with privacy sensitive regions.

Brennan Brodt and Alyssa Pierson are with the Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA [brodt, pierson]@bu.edu.

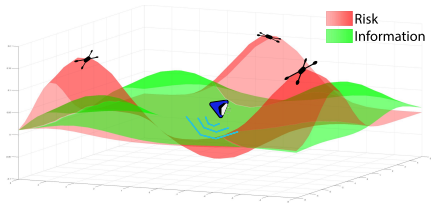


Fig. 1: Our agent (blue triangle) must gather information (green surface) over an environment, while avoiding external threats. Here, the external threats are illustrated as adversarial drones, which generate a risk field (red surface).

We formulate our risk-aware path-planning problem as a multi-objective optimization, which allows us to evaluate the trade-off between task completion and risk avoidance at each time step. We propose a sampling-based planner to generate Pareto-dominant solutions using local information in a receding horizon control policy. The agent selects the best trajectory through a dynamic weighted sum of objectives. Our approach provides locally-sampled, near-optimal solutions to the risk-aware path-planning problem. The key contributions of this work are as follows:

1) A multi-objective formulation of the risk-aware information gathering problem;
2) Dynamic weighting of objective functions to adaptively select solutions from a Pareto-dominant set;
3) Two local sampling based algorithms for solving the multi-objective optimization path planning problem.

The remainder of this paper is organized as follows: Section I-A presents related work. Section II formalizes the problem formulation and Section III describes our solution method. Section IV analyzes our simulation results and Section V state our conclusions.

A. Related Work

Our approach builds upon a rich foundation of work in risk-aware planning for robotic systems, using a multi-objective formulation to combine these methods with persistent monitoring for information gathering.

In path planning, risk is used to model modes of failure for robots. A classic example of risk aware planning in robotics research is obstacle avoidance, where the mode of failure is collision [2], [4], [14]. Other variations on risk aware planning include risk of sensor failure [3], risk of injuring people or damaging infrastructure [15], and risk of violating privacy [16], [17]. Early methods of risk aware planning rely on the offline development of models for environmental risk to plan with [18], [19]. More recent methods have used Markov processes [20] and trained models to estimate risk [21]. However, these methods require accurate a priori knowledge of the risk, which is not always available.

Data harvesting problems, where agents must traverse an environment to collect information, represent a broad class of robotics problems. With loose definitions for information and environments, these problems have an equally broad solution space. From offline methods like reinforcement learning [22] to online sampling methods [23], many robotics applications can be represented as data harvesting problems. An extension of data harvesting is persistent monitoring, where agents instead need to continuously harvest environmental data [5]. These problems are useful for tracking [6], mapping [24], and field estimation tasks [8]. Although frequently solutions to these problems must be found offline [25], there has been some work to suggest that receding horizon methods are effective in satisfying persistent monitoring tasks [7].

Multi-objective optimization methods are becoming more widespread in robotics problems, as modern tasks frequently require multiple specifications to be successful, which can even contradict each other [26], [27]. Multi-objective problems often require techniques which utilize careful user tuning and rarely yield unique solutions [28], [29]. Further, many of these techniques are computationally expensive, such as genetic algorithms, which can rarely be implemented in online path planning [30], [31]. However, recent work suggests that Pareto-optimality can be leveraged with modern controls, such as receding horizon [32] and ergodic search [33], to efficiently solve complex multi-objective robotics problems. In this paper, we pursue this concept of utilizing Pareto-optimality in conjunction with a sampling-based planner to generate trajectories online that satisfy both the information gathering and risk-aware path planning problems.

Autonomous Wildlife Monitoring: The method in this paper is motivated by autonomous wildlife monitoring. Ecological surveying is important for tracking and preserving global wildlife populations and there is recent work that suggest autonomous agents can play a major role in assisting with these operations [9], [34], [35], [36]. In fact, these surveys have already been carried out with a variety of animals including elephants [37], land mammals [11], and birds [10], [13], [38]. However, wild animals can have adverse reactions when in close proximity to autonomous agents [12], [35] but with careful planning we can reduce these reactions [10], [11]. With our method, environmental information can be used to direct agents carrying out a survey and risk can be used to model different species tolerances to their presence. In this way we can assist in ecological surveying while simultaneously respecting concerns regarding wildlife interactions with robots.

II. Problem Formulation

Our goal is to find trajectories for an agent such that we maximize the total information gathered while minimizing the accumulation of risk. Consider an unbounded environment $Q \subset \mathcal{R}^2$ with points in the environment denoted $q \in Q$. We denote the position of our agent $p(t) \in Q$, and assume integrator dynamics, $\dot{p} = u(t)$, where $u(t)$ is the desired control input that is bounded by, $|u(t)| \leq u_{\max}$. The agent is tasked with traversing an environment, gathering information, denoted by $\phi(q,t)$, while simultaneously avoiding risk, denoted by $\Gamma(q,t)$. In this section, we introduce environmental information and the dynamics that allow our agent to interact with it. Next, we define risk and quantify how the agent is expected to manage it. We then state our multi-objective formulation in Section III.

A. Information Dynamics and Gathering Information

We formulate our information gathering task with a data harvesting approach. We assume the total available information to collect in the environment, $\phi(q,t)$ comprises a sum of functions $f_i(q,t)$ representing individual goals or targets,

$$\phi(q,t) = \sum_{i=1}^{n} f_i(q,t), \tag{1}$$

where n is the total number of functions and $f_i(\cdot)$ models a particular point of interest. This construction allows for the encoding of multiple points of interest, as well as dynamic targets that relocate over time. Continuing with our wildlife monitoring example, each $f_i(\cdot)$ might represent a distinct nest location for a drone to visit and photograph. Within this paper, we represent $f_i(\cdot)$ as Gaussian functions, such that

$$\begin{aligned}\phi(q,t) &= \sum_{i=1}^{n} A_i \mathcal{N}_i(\mu_i(t), \Sigma_i(t)) \\ &= \sum_{i=1}^{n} \frac{A_i e^{-\frac{1}{2}(q-\mu_i(t))^T \Sigma_i(t)^{-1}(q-\mu_i(t))}}{2\pi|\Sigma_i|},\end{aligned} \tag{2}$$

where $0 < A_i < A_{\max}$ is a finite positive value and mean $\mu_i(t)$ and covariance $\Sigma_i(t)$ are allowed to vary with time. The choice of A_i is important in this setting because it serves to quantify the mass of information which is stored at each data harvesting location defined by mean μ_i.

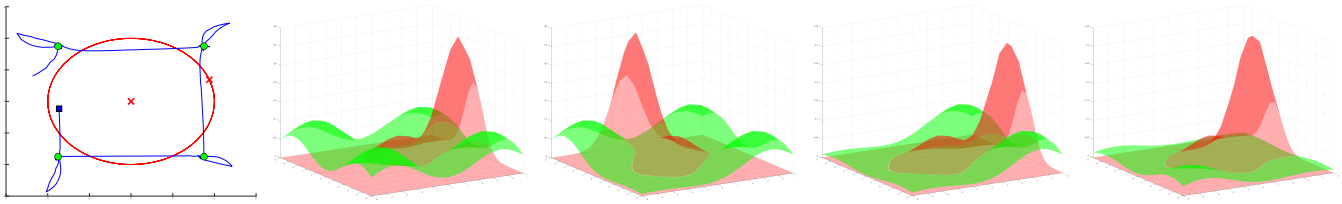
The information depletes over time based on the proximity of the agent, modeling the gathering dynamics. We assume the agent has some collection footprint, r_c. To gather information for a particular peak, the mean must fall within the collection footprint, or $\|p(t) - \mu_i(t)\| \leq r_c$. We define D_i to model the data collection from peak i by the agent,

$$D_i(t) = \begin{cases} \frac{\eta A_i}{1+||p(t)-\mu_i(t)||}, & \text{if } \|p(t) - \mu_i(t)\| \leq r_c \\ 0, & \text{else} \end{cases},$$

where $\eta \in [0,1]$ is the agent's data collection efficiency. The information gathered by an agent should also deplete the function $\phi(q,t)$, which we account for by changing the amplitude of individual peaks A_i. Further, we wish to allow for information to replenish in the system, relevant to persistent monitoring tasks. We define the overall dynamics of each information peak through $\dot{A}_i$,

$$\dot{A}_i = \begin{cases} \frac{-\eta A_i}{1+||p-\mu_i||} & \text{if } ||p - \mu_i|| \leq r_c \\ -c\tanh(\frac{A_{\max}}{A_i - A_{\max}}) + c & \text{else} \end{cases}, \tag{3}$$

where c tunes the gain of the saturation function when a peak is outside of the robot's collection radius. Note if the information does not replenish, then $c = 0$ creates a



(a) Environment (b) Fields at $t = 0$ (c) Fields at $t = 15$ (d) Fields at $t = 50$ (e) Fields at $t = 120$

Fig. 2: (a) Depicts an aerial view of the simulation environment, including sample trajectories for the agent and risk. (b)-(d) Depict the information fields (green surface) and environmental risk (red surface) as the simulation progresses.

depletion-only dynamic. The chosen dynamics in (3) drive the evolution of $\phi(q,t)$ in (2).

From the dynamics of how information evolves over time, we can write the information gathering task for our agent. We model this task as a maximization of the total accumulated information across all peaks over discretized time steps,

$$\max_{u} \mathcal{J} = \sum_{k=1}^{\tau} \sum_{i=1}^{n} D_i(t_k, p_k), \tag{4}$$

where t_k is the discretized time through present time $T = \tau\Delta t$, $u = [u_1^T, u_2^T, ..., u_\tau^T]^T$ is the control sequence at t_k, p_k is the position at t_k, and $s = [p_1^T, ..., p_\tau^T]^T$ is the trajectory of the agent. Next, we define the risk accumulation dynamics for the agent and introduce our risk-avoidance objective. We combine these policies in the multi-objective planning framework introduced in Section III.

B. Risk Dynamics and Risk Avoidance

Across the environment, different regions may carry risk to our agent. Similar to our construction of information density, we define a risk density that maps risk to specific locations within the environment. Intuitively, a larger value of risk indicates increased threat to the agent, as well as regions to avoid, while a lower value may indicate a safe region. We represent the overall risk $\Gamma(q,t)$ as a summation of m finite-valued functions $g_j(q,t)$ for $j = \{1, ..., m\}$,

$$\Gamma(q,t) = \sum_{j=1}^{m} g_j(q,t), \tag{5}$$

where each function $g_j(\cdot)$ represents the location of an external threat, adversary, or risky location. Here, we approximate each of the functions $g_j(\cdot)$ with Gaussian peaks defined mean $\mu_j(t)$, covariance $\Sigma_j(t)$, and amplitude B_j.

$$\Gamma(q,t) = \sum_{j=1}^{m} B_j \mathcal{N}(\mu_j(t), \Sigma_j(t)). \tag{6}$$

In our simulations, we present dynamic risk by ascribing motion to each of the peaks, which is realized through varying $\mu_j(t)$ while holding Σ_j constant.

We now present our model of how the agent interacts with risk. While it is important to avoid large values of risk at any instance in time, we also consider risk accumulation over time. For example, this risk function could approximate the ability of an adversary to make observations of our agent. With sufficient accumulated observations, our agent could be compromised. We consider risk accumulation over a recent time history, with a discounted sum of risk accumulated along a horizon. Intuitively, taking risky actions in the present should have should have a larger impact on planning than actions taken historically. Our second risk-avoidance objective is to choose actions that minimize the accumulated risk. We write this risk-avoidance objective function as

$$\min_{u} \mathcal{R} = \sum_{k=1}^{\tau} \gamma^{\tau-k} \Gamma(p_k, t_k), \tag{7}$$

where $\gamma \in [0,1]$ is a discount factor raised to the exponential depending on the time step and u is the control sequence for our agent over time. Figure 2 illustrates a scenario with four information peaks (green) and two risk peaks (red), which is further explained in Section IV.

III. Sampling Based Multi-Objective Path Planning

In Section II, we introduced our dynamics for information and risk, as well as individual objectives on gathering information and avoiding risk accumulation. Here, we present our multi-objective optimization problem to simultaneously solve for the control policy that achieves our two objectives, as well as our algorithm used in the simulations presented in Section IV. We formulate a receding horizon policy over some horizon T comprising τ time steps, $T = \tau\Delta t$. Over a fixed time horizon, optimal solutions may exist for known information and risk, however, solving for these becomes computationally intractable as the time horizon increases or with increasing complexity in $\phi(q,t)$ and $\Gamma(q,t)$. Our receding horizon controller evaluates local candidate solutions with the objective functions at each time step, then compares the set of candidates for Pareto-dominance [30] before using a weighted sum of the objectives to select a control input.

We first explain how Pareto-dominance extracts good candidate solutions to the multi-objective problem. Then, we describe how a weighted sum of objectives allows the agent to adaptively select the best solution from the set of Pareto-dominant solutions. Finally, we introduce two algorithms capable of generating and evaluating candidate solutions.

A. Pareto-Dominance of Candidate Trajectories

A key challenge in multi-objective optimization is the comparison of results across the multiple objective functions. Pareto-optimality provides a method for analyzing and ranking sets of solutions to multi-objective optimization problems

regardless of numerical scaling. Suppose we generate a set of candidate solutions by fixing control sequences across the time horizon τ, defined as $s^C = [{p_1^C}^T, ..., {p_\tau^C}^T]^T$ with objectives $\mathcal{J}^C(s^C)$ and $\mathcal{R}^C(s^C)$ along the trajectory,

$$\mathcal{J}^C(s^C) = \sum_{k=1}^{\tau} \sum_{i=1}^{n} D_i(t_k, p_k^C), \tag{8}$$

$$\mathcal{R}^C(s^C) = \sum_{k=\tau-H}^{\tau} \gamma^{\tau-k} \Gamma(t_k, p_k^C), \tag{9}$$

where $H \geq \tau$ represents an expanded horizon for evaluating risk, which incorporates both risk history as well as a projected accumulation over the horizon. Then, given we are minimizing $\mathcal{R}^{\mathcal{C}}$ and maximizing J^C, we define Pareto-dominated solutions as $s^d = [{p_1^d}^T, ..., {p_\tau^d}^T]^T$ such that,

$$\mathcal{J}^C(s^C) \geq \mathcal{J}^C(s^d) \wedge \mathcal{R}^C(s^C) < \mathcal{R}^C(s^d), \tag{10}$$

or,

$$\mathcal{R}^C(s^C) \leq \mathcal{R}^C(s^d) \wedge \mathcal{J}^C(s^C) > \mathcal{J}^C(s^d), \tag{11}$$

for at least one other candidate trajectory s^C. We then define Pareto-optimal trajectories as the set of non-dominated candidate solutions. Intuitively, this implies that any s^C is Pareto-optimal if no other candidate yields less accumulated risk and more gathered information. We then reject all dominated solutions as there exists an optimal solution that is guaranteed to provide better or equal performance on both objectives to any dominated solution.

B. Adaptive Weighting for Solution Selection

Selecting a solution from a set of dominant candidates is a common problem in multi-objective optimization. Traditionally, the solutions would be selected by the problem designer or through constraining the maximum or minimum allowable values on specific objective functions [28]. More recently, fuzzy logic has been used with good success in transforming a set of rules into a selection of solution [39]. However, these approaches have shortcomings in autonomous deployments, where we may not have operator feedback to select candidate solutions or design rules. We choose a method similar to [32], which uses a weighted sum to select the optimal solution from the candidates. Note we compute the values of $\mathcal{J}$ (8) and $\mathcal{R}$ (9) when generating our Pareto front of solutions. We create a combined multi-objective function

$$\max_u \mathcal{O} = \alpha \mathcal{J} - (1-\alpha)\mathcal{R}, \tag{12}$$

where $\alpha \in [0, 1]$ is a weighting that quantifies the relative importance between gathering information and avoiding risk. The choice of α will impact the overall behavior of the agent. At its limits, we note this reduces the problem to ignoring one of the objectives. We choose a dynamic $\alpha(t)$, which adapts to changes within the environment, to automatically select the best solution from the Pareto-optimal set. Intuitively, when we are collecting information in a low-risk region, we want to behave more greedily with respect to the information gathering task. Conversely, if we have accumulated a large amount of risk it is imperative to prioritize risk avoidance and prevent a mission failure. Thus, the desired value of $\alpha(t)$ depends on the current accumulated risk.

Since we consider the discounted sum of risk over our horizon, we can compute the upper bound on total risk accumulated by, $\mathcal{R}_{\max}$,

$$\mathcal{R}_{\max} \leq \sum_{k=0}^{H} \gamma^k \max(\Gamma(q, t_k)). \tag{13}$$

Note that this bound $\mathcal{R}_{\max}$ is not the same as a risk tolerance of the system. To relate this maximum accumulation to risk tolerance, we select an $\epsilon_{\max} \in [0, 1]$, which allows us to bound allowable risk accumulation. If $\mathcal{R}$ exceeds $\epsilon_{\max}\mathcal{R}_{\max}$, we would like our agent to prioritize risk aversion. Similarly, we introduce $\epsilon_{\min}$ as an activation threshold. This allows us to create a dynamic α that adapts to the accumulated risk,

$$\alpha = \begin{cases} 1 & \text{if } \mathcal{R}(t) \leq \epsilon_{\min}\mathcal{R}_{\max} \\ 0 & \text{if } \mathcal{R}(t) \geq \epsilon_{\max}\mathcal{R}_{\max} \\ e^{-\mathcal{R}+\epsilon_{\min}\mathcal{R}} & \text{else} \end{cases}. \tag{14}$$

From our adaptive $\alpha(t)$, we can demonstrate that the total accumulated risk for the agent also remains bounded by our designed threshold. To assert this bound, we assume the following on the properties of our risk function:

Assumption 1 (Risk Convexity)**.** *The risk function $\Gamma(q, t)$ is locally convex within the reachable set of the agent.*

Assumption 2 (Risk Dynamics)**.** *For a dynamic risk function defined in* (6)*, the velocity of any peak, $|\dot{\mu}_j| \leq \dot{p}$.*

In practice, Assumption 1 is valid for risk defined by (6) when the peaks are distributed in an environment. Assumption 2 implies that no peak within the risk function can move faster than our agents' velocity. Under these assumptions, Proposition 1 bounds the agent's accumulated risk.

Proposition 1. *For a multi-objective optimization in* (12)*, risk $\Gamma(q, t)$ in* (6)*, and adaptive α given by* (14)*, the agent's accumulated risk is bounded by*

$$\mathcal{R} \leq \epsilon_{\max} \sum_{k=0}^{H} \gamma^k \max(\Gamma(q, t)). \tag{15}$$

Proof. Consider some trajectory that yields an accumulated risk at the threshold of $\epsilon_{\max}\mathcal{R}_{\max}$. Upon reaching this threshold, $\alpha = 0$ by (14) and (12) becomes

$$\max_u \mathcal{O} = -\mathcal{R}, \tag{16}$$

which is a single objective optimization with respect to risk. Under Assumption 1, the risk is locally-convex over the reachable set of the agent's planning horizon, which implies the agent can follow a gradient descent to minimize its accumulated risk. Further, while the risk may be dynamic, Assumption 2 claims that the agent can move faster than the peak, and therefore any step following the gradient will continue to decrease the risk. Thus, the risk remains $\mathcal{R} \leq \epsilon_{\max}\mathcal{R}_{\max}$. Substituting our expression (13) yields (15), thus completing the proof. □

By Proposition 1, our agent maintains a risk under an allowable threshold during its deployment. Next, we present our sampling algorithm to generate candidate trajectories that solve our multi-objective problem over a given time horizon.

C. Sampling-Based Trajectory Generation

Generating candidate trajectories through sampling ensures a sufficient set of solutions to be evaluated for Pareto-dominance at each time step. We propose a type of Tree Search (TS) algorithm, inspired by dynamic programming [40] and the Rapidly-Exploring Random Tree algorithm [41], that allows us to quickly identify near-optimal solutions along the Pareto front of our multi-objective problem in (12). Further, this algorithm is able to plan using only local information. We first present Local Multi-Objective Tree Search (LMO-TS), which uses dense sampling to generate a large set of candidate trajectories in Algorithm 1, and then present Noisy Multi-Objective Gradient Descent Tree Search (nMOGD-TS) in Algorithm 2, which generates fewer candidate trajectories informed by local gradients.

For our LMO-TS detailed in Algorithm 1, from an initial position $p(t_0)$, we generate a set of b feasible $p(t_1)$ points. This repeats until we have b^τ nodes representing possible trajectories over time horizon τ. These trajectories are sorted by Pareto-dominance based on computed costs, then α is updated, which allows us to compute the multi-objective (12) and choose the best trajectory. This process iterates until the agent gathers all information within the environment.

While effective in some applications, Algorithm 1 suffers from the curse of dimensionality, as computational complexity explodes with larger planning horizons. Further, as a zero-order optimization technique, we do not leverage our current knowledge about the environment to improve our decisions. Finally, since the future positions are generated randomly, we require a large number of samples, b, to ensure our risk does not exceed the maximum threshold.

Alternatively, we propose nMOGD-TS, presented in Algorithm 2, whereby candidate nodes are generated using a noisy gradient descent on our objective functions. From an initial position $p(t_0)$, candidate nodes for $p(t_1)$ are generated in the directions of $\hat{\nabla}\phi$ and $\hat{\nabla}\Gamma$, followed by $b-2$ additional random candidate nodes. In practice, we find that forcing exploration along the gradients yields strong performance and can reduce the total number of b samples required.

Algorithm 1 Local Multi-Objective TS

Input: Initial Agent Position $p(t_0)$
Generate b random feasible $p(t_1)$
for $t_k \in [1, ..., \tau]$ **do**
  for $p_l(t_k) \in [1, ..., b]$ **do**
    Generate b random feasible $p(t_{k+1})$
Compute $\mathcal{J}$, $\mathcal{R}$ $\forall$ $s^C = [p^C(t_0)^T, ..., p^C(t_\tau)^T]^T$ ▷ (8,9)
Evaluate Pareto-dominance for all s^C
Eliminate all dominated sequences
Compute α ▷ (14)
Compute cost $\mathcal{O}$ for remaining s^C ▷ (12)
Select s^* s.t. $\mathcal{O}(s^*) \geq \mathcal{O}(s^C) \forall s^* \neq s^C$
Apply $u : p_0 \rightarrow p_1$

Algorithm 2 Noisy Multi-Objective Gradient Descent TS

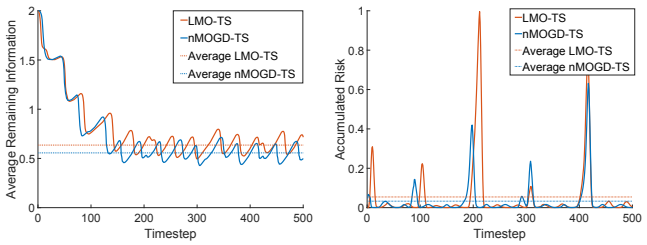
Input: Initial Agent Position $p(t_0)$
Sample Environment for $\phi(p(t_0), t_0), \Gamma(p(t_0), t_0)$ and $\phi(p(t_0) + \delta, t_0), \Gamma(p(t_0) + \delta, t_0)$
Approximate environmental gradients $\nabla\phi(t_0), \nabla\Gamma(t_0)$
Generate one $p(t_1)$ along $\nabla\phi$ and one along $\nabla\Gamma$, then generate $b-2$ additional random $p(t_1)$
for $t_k \in [1, ..., \tau]$ **do**
  for $p_l(t_k) \in [1, ..., b]$ **do**
    Sample Environment for $\phi(p(t_k), t_0)$, $\Gamma(p(t_k), t_0)$ and $\phi(p(t_k) + \delta, t_0)$, $\Gamma(p(t_k) + \delta, t_0)$
    Approximate Information and Risk Gradients $\hat{\nabla}\phi(t_k), \hat{\nabla}\Gamma(t_k)$
    Generate one $p(t_k)$ along $\hat{\nabla}\phi(t_k)$, one along $\hat{\nabla}\Gamma(t_k)$, and $b-2$ additional random $p(t_k)$
Compute $\mathcal{J}$, $\mathcal{R}$ $\forall$ $s^C = [p^C(t_0)^T, ..., p^C(t_\tau)^T]^T$ ▷ (8,9)
Evaluate Pareto-dominance for all s^C
Eliminate all dominated sequences
Compute α ▷ (14)
Compute cost $\mathcal{O}$ for remaining s^C ▷ (12)
Select s^* s.t. $\mathcal{O}(s^*) \geq \mathcal{O}(s^C) \forall s^* \neq s^C$
Apply $u : p_0 \rightarrow p_1$

IV. Results

MATLAB implementations of Algorithms 1 and 2 demonstrate the effectiveness of our approach. We compare the performance of our LMO-TS (Algorithm 1) with our nMOGD-TS (Algorithm 2). Our first scenario explores a persistent monitoring scenario, where the information replenishes when the agent is not nearby. Next, we remove the replenishment dynamics for a strict information gathering task and look at performance across 1000 trials. For both scenarios, we generate environments containing four distinct information peaks and two risk peaks. The information peaks, which might represent locations of wildlife, are modeled as time-invariant Gaussian peaks generated in the four corners of the environment. In Scenario 1, we generate one static risk peak in the center of the environment, and one dynamic risk peak that cycles among the information peaks, as drawn in Figure 2(a). For batch simulations in Scenario 2, we randomly generate the four peaks across the four corners, and randomly generate two dynamic risk peaks that follow elliptical orbits, as illustrated in Figure 4.

A. Scenario 1: Persistent Information Gathering

In this scenario, our goal is to demonstrate performance over a persistent information gathering task for the environment illustrated in Figure 2(a). We assume the agent's footprint is small relative to the scale of the environment, such that it can only gather information from one peak at a time. In turn, the three other peaks will replenish their information. Figure 3(a) plots the maximum amplitude of the information in the environment over time for both the



(a) Information over time (b) Accumulated Risk

Fig. 3: (a) Remaining environmental information for both implementations and steady-state averages. (b) Risk accumulation and average risk.

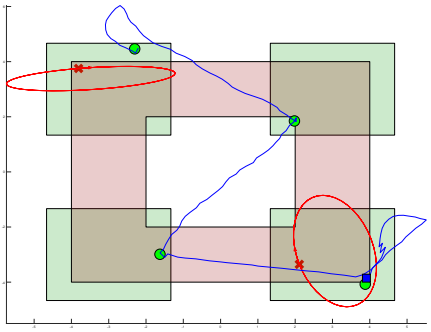
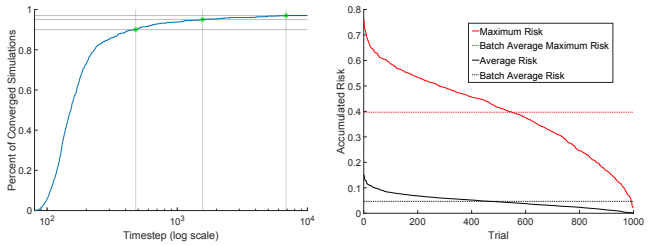


Fig. 4: Example of a randomly generated environment. Peaks (green circles) are generated within the green rectangles, and risk ellipses (red) are generated within the red region. The agent trajectory generated by Algorithm 2 is shown in blue.

Algorithm 1 (LMO-TS) and Algorithm 2 (nMOGD-TS). We demonstrate that we can maintain steady-state performance. Figure 3(b) plots the accumulated risk for both approaches over time. For this environment, $\max(\Gamma) = 2/\sqrt{(2\pi)}$, the planning horizon is $\tau = 3$, risk horizon $H = 2\tau$, decay rate $\gamma = 0.9$, and switching threshold of $\epsilon_{\max} = 0.9$ yields a maximum possible accumulated risk of 3.37 (15). We set the sampling quality $b = 10$ for Algorithm 1, and $b = 2$ for Algorithm 2. Using the sampling-dense approach, the risk stays well below the theoretical maximum, with an average risk of 0.054 over time and maximum risk of 0.996. Using nMOGD-TS, the average risk was 0.032 and maximum risk of 0.631, yielding a nearly 37% improvement despite fewer samples. This illustrates that utilizing gradient information to generate candidate trajectories can outperform a dense sampling method. However, a high sampling density might still be desirable in environments where it is impractical to compute the gradients.

B. Scenario 2: Randomized Gathering Tasks

To test the robustness of our approach, we generated batch simulations over randomized environments, illustrated in Figure 4. Further, we remove the assumption that the velocity of the risk peaks is slower than the agent. For each environment, we initialize four static information peaks, generated within the green rectangles of Figure 4. We generate the two random risk peaks within the red inset, where each risk peak follows a randomized elliptical trajectory. Note that



(a) Percent of Converged Simulations (b) Maximum and Average Accumulated Risk

Fig. 5: (a) Log plot of the completion rate of gathering all information for a given time step. (b) Sorted maximum and average values of accumulated risk for each simulation.

these orbits may contain or block an information peak, which we allow to rigorously test our algorithm. The information does not replenish, therefore, we consider the information gathering task complete once all information is collected from the environment.

Figure 5(a) depicts the completion rate of simulations versus the log of the number of iterations over 1000 trials of our nMOGD-TS approach. Overall, 970 trials successfully completed the information gathering task, with 900 trials completing before 600 time steps, and the last trial completing around 6000 time steps. Within the 30 incomplete trials, 26 were successful in harvesting three of the information peaks, 3 harvested two peaks, and only 1 trial had a harvest of one peak. Analysis into these trials indicated that the random generation of peaks either blocked the agent from moving past a peak, or prevented an agent from approaching the final peaks. However, a success rate of 97% demonstrates that our approach is quite adaptable to randomized configurations. Figure 5(b) plots the maximum and average accumulated risk over all trials, sorted in descending order. Each risk peak was a uniform Gaussian, with $\max(\Gamma) = 1/\sqrt{(2\pi)}$. We set the planning horizon to $\tau = 6$, risk horizon $H = 2\tau$, and decay rate of $\gamma = 0.9$. From these parameters, the theoretical maximum risk is 1.87 assuming no interference between peaks. As shown in Figure 5(b), the maximum accumulated risk was 0.761, with a mean maximum risk accumulation of 0.397 across all trials, and a mean average risk accumulation of 0.047. Thus, we see our approach successfully minimizes the risk accumulation while also accomplishing the information gathering task.

V. Conclusions

This paper proposes a new framework for risk-aware information gathering tasks. We considering risk as an additional objective to be optimized as opposed to a simple constraint, which allows for more flexibility in the decision making of an autonomous agent. To construct these types of problems, we propose a general form for modelling and quantifying environmental risks and information. Our algorithm efficiently solves the multi-objective optimization using locally-sampled information to generate Pareto-optimal trajectories. By defining dynamics for the decision making process, we have enabled an agent to complete complex specifications under high levels of uncertainty or danger.

REFERENCES

[1] R. Aggarwal, A. Soderlund, M. Kumar, and D. J. Grymin, "Risk-aware path planning for unmanned aerial systems in a spreading wildfire," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 9, pp. 1692–1708, 2022.

[2] J. Chu, F. Zhao, S. Bakshi, Z. Yan, and D. Chen, "Risk-aware path planning with uncertain human interactions," in *2021 American Control Conference (ACC)*, 2021, pp. 4225–4230.

[3] S. Mayya, R. K. Ramachandran, L. Zhou, V. Senthil, D. Thakur, G. S. Sukhatme, and V. Kumar, "Adaptive and risk-aware target tracking for robot teams with heterogeneous sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5615–5622, 2022.

[4] S. Safaoui, B. J. Gravell, V. Renganathan, and T. H. Summers, "Risk-averse rrt* planning with nonlinear steering and tracking controllers for nonlinear robotic systems under uncertainty," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 3681–3688.

[5] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2012.

[6] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "Multiagent persistent monitoring of targets with uncertain states," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 3997–4012, 2022.

[7] S. Welikala and C. G. Cassandras, "Event-driven receding horizon control for distributed persistent monitoring in network systems," *Automatica*, vol. 127, p. 109519, 2021.

[8] X. Lan and M. Schwager, "Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016.

[9] J. Linchant, J. Lisein, J. Semeki, P. Lejeune, and C. Vermeulen, "Are unmanned aircraft systems (uass) the future of wildlife monitoring? a review of accomplishments and challenges," *Mammal Review*, vol. 45, no. 4, pp. 239–252, 2015.

[10] E. Vas, A. Lescroël, O. Duriez, G. Boguszewski, and D. Grémillet, "Approaching birds with drones: first experiments and ethical guidelines," *Biology Letters*, vol. 11, no. 2, 2015.

[11] M. vanVuuren, R. vanVuuren, L. M. Silverberg, J. Manning, K. Pacifici, W. Dorgeloh, and J. Campbell, "Ungulate responses and habituation to unmanned aerial vehicles in africa's savanna," *PLOS ONE*, vol. 18, no. 7, pp. 1–19, 07 2023.

[12] M. A. Ditmer, J. B. Vincent, L. K. Werden, J. C. Tanner, T. G. Laske, P. A. Iaizzo, D. L. Garshelis, and J. R. Fieberg, "Bears show a physiological but limited behavioral response to unmanned aerial vehicles," *Current Biology*, vol. 25, no. 17, pp. 2278–2283, 2015.

[13] E. Brisson-Curadeau, D. Bird, C. Burke, D. A. Fifield, P. Pace, R. B. Sherley, and K. H. Elliott, "Seabird species vary in behavioural response to drone census," *Scientific reports*, vol. 7, no. 1, pp. 17 884–9, 2017.

[14] K. Cai, C. Wang, S. Song, H. Chen, and M. Q.-H. Meng, "Risk-aware path planning under uncertainty in dynamic environments," *Journal of intelligent & robotic systems*, vol. 101, no. 3, 2021.

[15] S. Primatesta, G. Guglieri, and A. Rizzo, "A risk-aware path planning strategy for uavs in urban environments," *Journal of intelligent & robotic systems*, vol. 95, no. 2, pp. 629–643, 2019.

[16] Y. Pan, S. Li, J. L. Chang, Y. Yan, S. Xu, Y. An, and T. Zhu, "An unmanned aerial vehicle navigation mechanism with preserving privacy," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[17] Y. Luo, Y. Yu, Z. Jin, Y. Li, Z. Ding, Y. Zhou, and Y. Liu, "Privacy-aware uav flights through self-configuring motion planning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1169–1175.

[18] G. S. Aoude, B. D. Luders, D. S. Levine, and J. P. How, "Threat-aware path planning in uncertain urban environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 6058–6063.

[19] A. A. Pereira, J. Binney, B. H. Jones, M. Ragan, and G. S. Sukhatme, "Toward risk aware mission planning for autonomous underwater vehicles," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3147–3153.

[20] S. Feyzabadi and S. Carpin, "Risk-aware path planning using hierarchical constrained markov decision processes," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, 2014, pp. 297–303.

[21] M. Ono, T. J. Fuchs, A. Steffy, M. Maimone, and J. Yen, "Risk-aware planetary rover operation: Autonomous terrain classification and path planning," in *2015 IEEE Aerospace Conference*, 2015, pp. 1–10.

[22] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Multi-uav path planning for wireless data harvesting with deep reinforcement learning," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1171–1187, 2021.

[23] M. Tzes, Y. Kantaros, and G. J. Pappas, "Distributed sampling-based planning for non-myopic active information gathering," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5872–5877.

[24] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple uavs for persistent surveillance: Algorithm and flight test results," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1236–1251, 2012.

[25] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "A semidefinite programming approach to discrete-time infinite horizon persistent monitoring," in *2021 European Control Conference (ECC)*, 2021, pp. 799–804.

[26] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770–784, 2007.

[27] B. Brodt and A. Pierson, "Obscuring objectives with pareto-optimal privacy-aware trajectories in multi-robot coverage," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7670–7676.

[28] K. Ellefsen, H. Lepikson, and J. Albiez, "Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures," *Applied Soft Computing*, vol. 61, pp. 264–282, 2017.

[29] D. He, Y. Zhang, and S. Yu, "Prioritized multi-objective model predictive control without terminal constraints and its applications to nonlinear processes," *Optimal Control Applications and Methods*, vol. 42, no. 4, pp. 1030–1044, 2021.

[30] D. E. D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass.: Addison-Wesley Publishing Company, 1989.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[32] K. Lee, S. Martínez, J. Cortés, R. H. Chen, and M. B. Milam, "Receding-horizon multi-objective optimization for disaster response," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 5304–5309.

[33] Z. Ren, A. K. Srinivasan, B. Vundurthy, I. Abraham, and H. Choset, "A pareto-optimal local optimization framework for multiobjective ergodic search," *IEEE Transactions on Robotics*, pp. 1–12, 2023.

[34] K. S. Christie, S. L. Gilbert, C. L. Brown, M. Hatfield, and L. Hanson, "Unmanned aircraft systems in wildlife research: current and future applications of a transformative technology," *Frontiers in Ecology and the Environment*, vol. 14, no. 5, pp. 241–251, 2016.

[35] L. Schad and J. Fischer, "Opportunities and risks in the use of drones for studying animal behaviour," *Methods in Ecology and Evolution*, vol. 14, no. 8, pp. 1864–1872, 2023.

[36] A. Delplanque, S. Foucher, P. Lejeune, J. Linchant, and J. Théau, "Multispecies detection and identification of african mammals in aerial imagery using convolutional neural networks," *Remote Sensing in Ecology and Conservation*, vol. 8, no. 2, pp. 166–179, 2022.

[37] C. Vermeulen, P. Lejeune, J. Lisein, P. Sawadogo, and P. Bouché, "Unmanned aerial survey of elephants," *PLOS ONE*, vol. 8, no. 2, pp. 1–7, 02 2013.

[38] A. E. McKellar, N. G. Shephard, and D. Chabot, "Dual visible-thermal camera approach facilitates drone surveys of colonial marshbirds," *Remote Sensing in Ecology and Conservation*, vol. 7, no. 2, pp. 214–226, 2021.

[39] J. J. Valera García, V. Gómez Garay, E. Irigoyen Gordo, F. Artaza Fano, and M. Larrea Sukia, "Intelligent multi-objective nonlinear model predictive control (imo-nmpc): Towards the 'on-line' optimization of highly complex control problems," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6527–6540, 2012.

[40] R. Bellman, *Dynamic programming*, ser. Rand Corporation research study. Princeton: Princeton University Press, 1957.

[41] S. M. LaValle and J. James J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.