Covert Planning against Imperfect Observers

Haoxiang Ma
University of Florida
Gainesville, FL, United States
hma2@ufl.edu

Chongyang Shi
University of Florida
Gainesville, FL, United States
c.shi@ufl.edu

Shuo Han
University of Illinois Chicago
Chicago, IL, United States
hanshuo@uic.edu

Michael R. Dorothy
DEVCOM Army Research Laboratory
Adelphi, MD, United States
michael.r.dorothy.civ@army.mil

Jie Fu
University of Florida
Gainesville, FL, United States
fujie@ufl.edu

ABSTRACT

Covert planning refers to a class of constrained planning problems where an agent aims to accomplish a task with minimal information leaked to a passive observer to avoid detection. However, existing methods of covert planning often consider deterministic environments or do not exploit the observer's imperfect information. This paper studies how covert planning can leverage the coupling of stochastic dynamics and the observer's imperfect observation to achieve optimal task performance without being detected. Specifically, we employ a Markov decision process to model the interaction between the agent and its stochastic environment, and a partial observation function to capture the leaked information to a passive observer. Assuming the observer employs hypothesis testing to detect if the observation deviates from a nominal policy, the covert planning agent aims to maximize the total discounted reward while keeping the probability of being detected as an adversary below a given threshold. We prove that finite-memory policies are more powerful than Markovian policies in covert planning. Then, we develop a primal-dual proximal policy gradient method with a two-time-scale update to compute a (locally) optimal covert policy. We demonstrate the effectiveness of our methods using a stochastic gridworld example. Our experimental results illustrate that the proposed method computes a policy that maximizes the adversary's expected reward without violating the detection constraint, and empirically demonstrates how the environmental noises can influence the performance of the covert policies.

KEYWORDS

Markov decision processes; Deception; Covert Planning

ACM Reference Format:
Haoxiang Ma, Chongyang Shi, Shuo Han, Michael R. Dorothy, and Jie Fu. 2024. Covert Planning against Imperfect Observers. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Covert planning refers to accomplishing some tasks with minimal information leaked to a passive observer to avoid detection. Such planning algorithms are useful in various security applications including surveillance and crime prevention. For example, a security patrolling agent (robot or human) would need to minimize the knowledge of his presence while collecting information in several regions of interest. In a contested search and rescue mission, a human-robot team may be sent to gather information in the hostile environment without being detected. Game AI is another area that uses covert planning algorithms [1]. An AI player can use covert planning to cause the element of surprise to the human player, thus making the game more entertaining and realistic.".

In this work, we study a class of covert planning in stochastic environments. Consider a planning problem in a stochastic environment modeled as a Markov Decision Process (MDP). The goal of the planning agent is to maximize a discounted total reward, while ensuring covert behavior against an observer. Specifically, the covert constraint requires that with a high probability, an observer could not detect any deviation from a nominal behavior modeled by a Markov chain, with its imperfect observation of the agent's path. We employ a sequential likelihood ratio test to construct the covert constraint and prove that a finite-memory policy can be more powerful than a Markovian policy in the resulting constrained MDP. Due to the intractable search space for finite-memory policies, we develop a primal-dual gradient-based policy search method to compute an optimal and covert Markov policy. To mitigate the distribution draft and improve the stability, we employ a two-time-scale approach and a sample-efficient estimation for the policy gradient. We demonstrate the performance of the proposed algorithms using a security patrolling agent tasked with visiting a set of goal states, while the nominal behavior is obtained from other users that perform routine activities in the dynamic environment.

Related Work. The work on covert path planning is closely related to stealthy evader strategies in the pursuit-evasion game and covert robots [2]. In a pursuit-evader game, a team of pursuers aims to locate and capture one or more evaders. The pursuit-evasion game can be of imperfect information where the evader hides at or moves between locations unobservable to the pursuer. The interactions are often formulated as a hide-and-seek game [9, 27]. The equilibrium of pursuit-evasion games with sensing limitation has been investigated for continuous-state dynamical systems [4, 10] and deterministic games on graphs [11]. Our formulation can be

viewed as planning for a stealthy evader who aims to achieve the goal while remaining hidden in a stochastic environment. Specifically, the nominal behavior in our setting is simply the environment dynamics without the presence of an evader. However, due to the stochastic dynamics and noisy observation, the covert planning cannot be solved with existing algorithms for hide-and-seek games or plan obfuscation that consider deterministic dynamics.

For stochastic systems modeled as MDPs, deceptive planning has been studied. In [13], the authors study a deceptive planning problem where a supervisor determines a reference policy for the agent to follow, while the agent instead uses a different, deceptive policy to achieve a secret task. The planning problem is formulated such that the agent minimizes the divergence between the distributions under the deceptive policy and those under the supervisor's policy, while ensuring the probability of achieving a secret task is greater than a given threshold. The KL-divergence between a policy and a reference policy is also used as a metric for deceptiveness in [21, 22] to deceive the supervisor/observer into believing the agent's policy follows a given reference. In [14], the authors study a similar problem of exploiting observation noises for deception and propose to include minimizing the KL divergence between the observation of a nominal policy and that of the deceptive agent's policy as an additional objective for deceptiveness. They show the planning problem is NP-hard and provide an approximate solution to find an optimal mixture policy (a policy defined as a weighted sum of basis policies). In [16, 18], the authors consider the observer construct a reward estimator. The deceptiveness is measured by the entropy in the estimator. [23] studies rewards related to exaggeration (pretending to reach a fake goal) or ambiguity of the goal. Other related work includes plan recognition [20, 25] where the observer aims to infer the goal of an agent, given a finite set of possible goals and its observation. Covert planning can be viewed as counter-plan recognition.

In comparison to deceptive planning in MDPs [13, 21, 22], our work differs in the following aspects: 1) Most existing work assumes full observations of the passive observer, while we consider observers with imperfect observations and address how the agent can leverage both the noise in the environmental dynamics and imperfect information of the observer for covertness. Due to partial observations, the detection constraint cannot be represented by a cumulative cost/reward. This explicit consideration of observation functions provides more insight into verifying the effectiveness of a sensor design against deceptive, covert planning adversaries. 2) In comparison to [14] which exploits the observation noises for deception, our covert constraint based on sequential likelihood ratio test introduces a chance-constrained MDP, rather than a multi-objective MDP or MDP with a constraint on KL divergence between two hidden Markov models. This chance-constraint allows the planning algorithm to explicitly bind the probability of detection. Further, we prove finite-memory policies could obtain a higher objective value than Markov policies for a fixed covert constraint. We develop policy gradient method to directly search the optimal covert policy in parameterized poicy space, instead of restricting to the class of mixture policies as [14] did.

This paper is organized as follows. We provide the preliminary definitions and formal problem statement in Section 2. In Section 3, we analyze the $(1-\alpha)$-covert planning problem, propose a two-time-scale primal-dual proximal policy gradient method, and compute a (locally) optimal covert policy. In Section 4, we demonstrate our results using a stochastic gridworld example. We show our framework maximizes the adversary's expected reward without violating the detection constraints. We conclude in Section 5 and discuss future directions.

2 PRELIMINARIES AND PROBLEM FORMULATION

Notations Let $\mathbf{R}$ denote the set of real numbers and $\mathbf{R}^n$ the set of real n-vectors. The vector of all ones is represented as $\mathbf{1}$. The notation z_i refers to the i-th component of a vector $z \in \mathbf{R}^n$ or to the i-th element of a sequence $z_1, z_2, \ldots$, which will be clarified by the context. The set of probability distributions over a finite set Z is denoted as $\mathcal{D}(Z)$.

The planning problem is modeled as a Markov decision process $M = (S, A, P, s_0, R, \gamma)$ where S is a finite set of states, A is a finite set of actions, $P : S \times A \to \mathcal{D}(S)$ is a probabilistic transition function and $P(s'|s, a)$ is the probability of reaching state s' given that action a is taken at the state s. The initial state is s_0. The planning objective for the agent (referred to as player 1/P1) is described by a reward function $R : S \times A \to \mathbf{R}$.

A policy for M is a function $\pi : \mathscr{D} \to \mathscr{C}$ where it is called *memoryless or Markovian* if $\mathscr{D} = S$; *finite-memory* if $\mathscr{D} = (S \times A)^* S$; *deterministic* if $\mathscr{C} = A$, and *randomized* if $\mathscr{C} = \mathcal{D}(A)$. For a Markovian policy $\pi : S \to \mathcal{D}(A)$, P1's value function $V^\pi : S \to \mathbf{R}$ is defined as

$$V^\pi(s) = E_\pi[\sum_{k=0}^{\infty} \gamma^k R(s_k, \pi(s_k))|s_0 = s],$$

where E_π is the expectation with respect to the probability distribution induced by the policy π from the MDP M, and s_k is the k-th state in the Markov chain induced from the MDP M under the policy π, starting from state s.

Given the MDP M, the goal of P1 is to maximize the total discounted reward given some discounting factor γ. In addition, the agent must ensure its behavior is covert with respect to a passive observer (player 2/P2) whose imperfect observation function is given as follows.

Definition 1 (Observation function of P2)**.** Let O be a finite set of observations. The state-observation function of P2 is $\text{Obs}_S : S \to \mathcal{D}(O)$ that maps a state s to a distribution $\text{Obs}_S(s)$ over observations. The action observation function is state-dependent, defined as $\text{Obs}_A : S \times A \to \mathcal{D}(O)$ that maps an action a from s to a distribution $\text{Obs}_A(s, a)$ over observations.

Considering the actions that are state-dependent allows for the most general class of observation functions. Without loss of generality, we denote $\text{Obs} : S \cup S \times A \to \mathcal{D}(O)$ as the combined state and action observation function.

The goal of P2 is to detect if there is any deviation in the agent's behavior from a normal user's behavior. A normal user follows a Markovian policy π_0 in M, referred to as the *nominal policy*. As a result, the normal user's behavior is modeled as a hidden Markov model induced from the original MDP M given some nominal user policy and the defender's observation function.

Definition 2 (HMM modeling the P2's observation given the nominal policy π_0)**.** Given the MDP $M = (S, A, P, s_0)$, the nominal policy $\pi_0 : S \to \mathcal{D}(A)$, and an observation function Obs : $S \cup S \times A \to \mathcal{D}(O)$, the stochastic process of observations is captured using a discrete HMM(with state emission),

$$M_0 = \langle S \cup S \times A, O, \mathbf{P}, \mathbf{E}, s_0 \rangle$$

- $S \cup S \times A$, including two types of states, a decision state s at which an action will be selected, and a nature's state (s, a) at which the next state will be determined according to a probability distribution.
- O is an alphabet, the set of observations;
- $\mathbf{P} : (S \cup S \times A) \times (S \cup S \times A) \to [0, 1]$ is the mapping defining the probability of each transition. The following constraints are satisfied: For $s \in S$,

  $$\mathbf{P}(s, (s, a)) = \pi_0(s, a);$$

  For $(s, a) \in S \times A$,

  $$\mathbf{P}((s, a), s') = P(s'|s, a);$$

- $\mathbf{E} : (S \cup S \times A) \times O \to [0, 1]$ is the mapping defining the emission probability of observation at a state that satisfies the following constraints:

  $$\mathbf{E}(s, o) = \text{Obs}(o \mid s),$$

  and

  $$\mathbf{E}((s, a), o) = \text{Obs}(o \mid s, a).$$

It can be validated that the probabilistic transition function and emission function are well-defined.

The covert-planning problem is informally stated as follows.

Definition 3. Given an MDP M and a nominal behavior modeled as an HMM M_0, compute a policy π for P1 that maximizes P1's total discounted reward, while ensuring, with a high probability $1 - \alpha$ for $\alpha \in (0, 1)$, P2 cannot detect P1's deviation from the nominal behavior M_0.

3 MAIN RESULTS

First, we show that a finite-memory policy can be more powerful than a Markov policy for being $(1 - \alpha)$-covert.

Theorem 1. Given a MDP M, P2's observation function Obs and reward function R, a Hidden Markov Model (HMM) M_0 modeling P2's observations given the nominal, Markovian policy π_0, and a parameter $\alpha \in (0, 1)$, there may exist an $(1 - \alpha)$-covert finite-memory optimal policy that maximizes the total discounted reward but no $(1 - \alpha)$-covert Markovian policy that can attain the same value of that finite-memory policy for the reward function.

Proof. We prove this theorem by constructing such a case. Consider a simple MDP with $S = \{1, 2\}$, $A = \{H, T\}$ and stochastic transitions $P(1|1, H) = 1$, $P(2|1, T) = 1$, $P(1|2, H) = 0.5$, $P(2|2, H) = 0.5$ and $P(2|2, T) = 1$, illustrated in Fig. 1.

An agent is allowed to take N actions and then stop. The nominal hypothesis is that at each state, the user selects "H" or "T" with equal probabilities. The observer can observe the actions, but not the states. It is not hard to see that the hidden Markov model (for observations) induced by the nominal policy describes the sequence of outcomes for flipping a fair coin for N times, in which the number of heads is a random variable that follows a binomial distribution. The reward function for the planning agent is $R(1, H, 1) = 1$, $R(2, H, 1) = 1$, and for other (state, action, state) pairs, the reward is 0. Therefore, if the agent is to maximize the total reward without covertness, it would select only action H at both states.

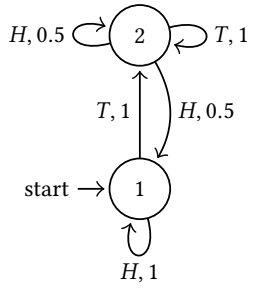

Figure 1: An MDP example

Consider the observer employing hypothesis testing: With a confidence level α and total N of actions, the observer would reject the nominal hypothesis if either the number of heads or the number of tails is greater than or equals an integer K. For example, if $\alpha = 0.05$, $N = 10$, then the null hypothesis is rejected if the number of heads is less than 2 or greater than 8.

Let ζ be the random variable that represents the number of times "H" is observed. Assuming $N = 2$ (the extension for $N \geq 2$ is discussed later), if the observer rejects the null hypothesis for observing more than one "H", then the covert constraint can be described as $Pr(\zeta \leq 1) \geq 1-\rho$, where $\rho \in (0, 1)$ is a given threshold.

First, we consider the initial state to be 1 and a Markovian policy π_2 defined by $\pi_2(H|1) = \alpha$, and $\pi_2(H|2) = \beta$ for $\alpha, \beta \in [0, 1]$. Considering all trajectories generated by taking 2 actions, which are {"111", "112", "121", "122"}, the corresponding probabilities of generating the trajectories are $\alpha^2, \alpha(1-\alpha), (1-\alpha)\frac{\beta}{2}, (1-\alpha)(1-\beta) + (1-\alpha)\frac{\beta}{2}$ respectively. In order to find a Markovian policy that satisfies the covert constraint, we have $\alpha^2 \leq \rho$. The corresponding value of the Markovian policy is $2\alpha^2 + \alpha(1-\alpha) + (1-\alpha)\frac{\beta}{2} = \alpha^2 + \alpha + (1-\alpha)\frac{\beta}{2}$. This value is smaller or equal to $\alpha^2 + \frac{\alpha+1}{2}$, the equality holds when $\beta = 1$.

Next, we consider a finite-memory policy $\pi_2^\dagger$, where the agent takes action "H" at step 1, and takes action "H" with probability ρ, action "T" with probability $1 - \rho$ at step 2. Note that this policy cannot be represented by a Markov policy. It is clear the finite-memory policy satisfies the covert constraint. And the corresponding reward is $1 + \rho$ in two steps.

Let r_1 denote the optimal Markovian policy's total reward, which is attained when $\beta = 1$, and r_2 denote the finite-memory policy's total reward. Then the following relation can be established:

$$r_2 - r_1 = 1 + \rho - \alpha^2 - \frac{\alpha + 1}{2} \tag{1}$$

$$\geq 1 - \frac{\alpha + 1}{2} \tag{2}$$

$$\geq \frac{1}{2}(1 - \sqrt{\rho}) \tag{3}$$

where the inequality is due to the covert constraint $\alpha^2 \leq \rho$ and thus $-\alpha \geq -\sqrt{\rho}$. Since $\rho \in (0,1)$, we have $r_2 - r_1 > 0$, which means the finite-memory covert policy performs better than the optimal Markovian covert policy when we take two actions.

Next, we extend the case to N actions for $N > 2$. Using the hypothesis testing for binomial distribution, the observer would reject the nominal hypothesis if either the number of heads or that of tails exceeds an integer K. First, let π_n denote the optimal Markovian policy that the agent can follow while satisfying the covert constraint. Let's construct a new policy π'_n such that for steps $1, 2, \ldots, N-2$, π'_n is exactly the same as π_n. At step $N-1$, if the agent is at state 2, then the agent keeps following policy π_n; if the agent is at state 1 and the number of "H" taken by the policy π_n till this step is X, the decision for the next two actions will consider two different cases: 1) If $X+2 < K$, then the covert constraint allows two more "H" actions and the agent follows a finite-memory policy that takes two "H" actions. In this case, any Markovian policy will have a reward at most the same as the finite-memory policy for the last two steps. 2) If $X+2 \geq K$ and $X < K$, then the covert constraint allows only 0 or 1 "H" actions, then it is the case we discussed previously when $N = 2$, the best Markovian policy the agent can follow is π_2 at the last 2 steps. However, the total reward obtained by following π_2 is smaller than the reward obtained by following the finite-memory policy $\pi_2^\dagger$ given $N = 2$. The newly constructed π'_n that commits to some finite memory policy in the last two steps is non-Markovian and satisfies the covert constraint. It also attains a better value than the optimal Markovian policy π_n that satisfies the covert constraint. Based on this constructed example, we conclude that a finite-memory policy can perform better than a Markovian policy when a covert constraint is enforced. □

Despite the theorem showing that finite-memory policy is more powerful, it is intractable to compute because both the structure (memory states and transitions) and the mapping from the memory states to distributions over actions are unknown. Thus, in the next section, we restrict our solution to Markovian policy space.

3.1 Computing a Covert Markovian Policy

First, we review the sequential likelihood ratio test [8] for hypothesis testing of hidden Markov models. Based on hypothesis testing, we then formalize the notion of $(1-\alpha)$-covert policy.

Notations: We introduce a set of parameterized Markovian policies $\{\pi_\theta \mid \theta \in \Theta\}$ where Θ is a finite-dimensional parameter space. For any Markovian policy π_θ parameterized by θ, the Markov chain induced by π_θ from the MDP M is denoted M_θ: $\{X_t, A_t, t \geq 0\}$ where X_t is the random variable for the t-th state and A_t is the random variable for the t-th action. For a run $x = s_0 a_0 s_1 a_1 \ldots s_n$, $P(x; M_\theta)$ is the probability of the run x in the Markov chain M_θ. We denote the distribution over a run by a random vector X with support $\mathcal{X}$. Each sample of X is a finite run x.

Softmax Parameterization. A natural class of policies is parameterized by the softmax function,

$$\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a' \in A} \exp(\theta_{s,a'})}. \tag{4}$$

The softmax has good analytical properties including completeness and differentiability. It can represent any stochastic policy. In this work, we consider the policy to be in the softmax parameterization form.

The observation distribution given policy π_θ is modeled by the HMM that can be constructed similar to M_0 in Def. 2. We denote the observation as a random vector Y with a support $\mathcal{Y}$. Each sample of Y is a finite observation sequence $y = o_0 o_1 \ldots o_n$. The probability of observing y given the policy π_θ is denoted $P(y; M_\theta)$. Likewise, $P(x; M_0)$(resp. $P(y; M_0)$) is the probability of a run x (resp. an observation y) in the nominal model M_0,

Let's consider the case with an *informed* defender who has access to the policy π_θ (the alternative hypothesis). Under the Markov policy π, the stochastic process $\{O_i, i \geq 0\}$ is a hidden Markov model where O_i is the random variable representing the observation at the i-th time step. Let $Y_{1:n} \coloneqq O_0, O_1, O_2, \ldots, O_n$ be a sequence of random variables for the finite observations with the unknown model M. The likelihood ratio is defined as:

$$S_n \coloneqq \frac{P(Y_{1:n}; M_\theta)}{P(Y_{1:n}; M_0)}.$$

The sequential probability ratio test (SPRT) of $M = M_0$ versus $M = M_\theta$ stops sampling at the stage

$$T \coloneqq \inf\{n : \log S_n \leq \epsilon \text{ or } \log S_n \geq \phi\},$$

where ϵ and ϕ are two parameters defined by SPRT and $\epsilon < \phi$. The test accepts the null hypothesis that $M = M_0$ if $\log S_T \leq \epsilon$, and accepts the alternative hypothesis $M = M_\theta$ if $\log S_T \geq \phi$. Here, the thresholds ϵ, ϕ are determined based on the bounds on Type I and Type II errors in SPRT for hidden Markov models [8].

Thus, given a single observation sequence y, the SPRT cannot reject the null hypothesis if

$$\log \frac{P(y; M_\theta)}{P(y; M_0)} \leq \epsilon.$$

Because each observation y is generated with probability $P(y; M_0)$, the following constraint enforces, with a probability greater than α, that the null hypothesis not rejected:

$$\Pr(\log \frac{P(Y; M_\theta)}{P(Y; M_0)} > \epsilon; M_\theta) \leq \alpha,$$

where $\Pr(E; M_\theta)$ is the probability of event E in the hidden Markov model M_θ. For convenience, we refer the term $\Pr(\log \frac{P(Y;M_\theta)}{P(Y;M_0)} > \epsilon; M_\theta)$ as the *detection probability* and $\log \frac{P(Y;M_\theta)}{P(Y;M_0)} > \epsilon$ as the detection condition.

Definition 4. Given a small constant $\alpha \in [0,1]$, a policy π_θ is $(1-\alpha)$-covert if it is the solution to the following problem.

$$\underset{\pi_\theta}{\text{maximize}} \quad V^{\pi_\theta}(s_0) \tag{5}$$

$$\text{s.t.} \quad \Pr\left(\log \frac{P(Y; M_\theta)}{P(Y; M_0)} > \epsilon; M_\theta\right) \leq \alpha, \tag{6}$$

where M_θ is the HMM induced by policy π_θ given the MDP M.

Remark 1. In Def. 4, we conservatively assume an informed observer who has access to the agent's policy. In reality, the observer is not informed and his detection performance can be worse than the informed observer. The assumption of an informed observer is also common in minimal information-leakage communication

channel design to ensure strong information security and privacy [15]. In our case, this assumption provides a strong guarantee for covertness.

3.2 Primal-dual Proximal Policy Gradient for Covert Optimal Planning

By the method of Lagrange multipliers [3], we can formulate the problem in (5) into an unconstrained max-min problem. For our problem, the Lagrangian function is given by

$$L(\theta, \lambda) = V(s_0, \theta) + \lambda \left(\alpha - \Pr\left(\log \frac{P(Y; M_\theta)}{P(Y; M_0)} > \epsilon; M_\theta \right) \right),$$

where θ is the primal variable and $\lambda \geq 0$ is the dual variable. Here, we replace the policy π_θ with the policy parameter θ for clarity and also rewrite $V^{\pi_\theta}(s_0)$ as $V(s_0, \theta)$. The original constrained optimization problem in (5) can be reformulated as

$$\underset{\theta}{\text{maximize}} \quad \min_{\lambda \geq 0} L(\theta, \lambda).$$

Because the value function can be a non-convex function of the policy parameters, we present a primal-dual gradient-based policy search method aiming to compute a (locally) optimal policy that satisfies the constraint. This method uses two-time-scale updates for the primal and dual variables. The primal variable θ is updated at a faster time scale, while the dual variable λ is updated at a slower time scale (one update after several primal updates).

When performing the gradient computation of L with respect to θ, it is observed that the constraint involves taking expectation over a distribution that depends on the decision variable θ. To mitigate the distribution shift [6] in stochastic optimization, we then introduce a proximal policy gradient to bound the distribution shift between two updates on the primal variable. That is, at the t-th iteration, when computing the gradient of $L(\theta, \lambda)$ with respect to θ, we include a KL-divergence between the trajectory distribution P_{θ_t} under the current policy parameterized by θ_t and the trajectory distribution P_θ under the new policy parameterized by θ, similar to the proximal policy optimization method [24]. After including the KL-divergence, the Lagrangian function becomes

$$L^\beta(\theta, \lambda) = V(s_0, \theta) + \lambda \left(\alpha - \Pr\left(\log \frac{P(Y; M_\theta)}{P(Y; M_0)} - \epsilon > 0; M_\theta \right) \right) - \beta D_{KL}(P_{\theta_t} || P_\theta),$$

where $\beta > 0$ is a scaling parameter that can be dynamically adjusted and

$$D_{KL}(P_{\theta_t} || P_\theta) = \mathbf{E}_{x \sim P_{\theta_t}} \left(\log \frac{P_\theta(x)}{P_{\theta_t}(x)} \right) = \sum_{x \in \mathcal{X}} P_{\theta_t}(x) \log(\frac{P_\theta(x)}{P_{\theta_t}(x)}),$$

where $P_\theta(x)$ (resp. P_{θ_t}) is the probability of the run x in the Markov chain M_θ (resp. M_{θ_t}).

Taking the gradient of $L^\beta(\theta, \lambda)$ with respect to θ, note that $\Pr\left(\log \frac{P(Y;M_\theta)}{P(Y;M_0)} > \epsilon; M_\theta \right) = \mathbf{E}_{y \sim M_\theta} \left[\mathbf{1}(\log \frac{P(y;M_\theta)}{P(y;M_0)} - \epsilon > 0) \right]$, where $\mathbf{1}(E)$ is the indicator function that evaluates to one if E is true and zero otherwise. We have

$$\nabla_\theta L^\beta(\theta, \lambda) = \lambda \nabla_\theta \left(\alpha - \mathbf{E}_{y \sim M_\theta} \left[\mathbf{1}(\log \frac{P(Y = y; M_\theta)}{P(Y = y; M_0)} - \epsilon > 0) \right] \right) + \nabla_\theta V(s_0, \theta) - \beta \nabla_\theta D_{KL}(P_{\theta_t} || P_\theta).$$

We approximate the gradient using samples generated from the current chain M_{θ_t}. Each sample is a pair (x_i, y_i) that includes: 1) a run $x_i = s_{i,0} a_{i,0} s_{i,1} a_{i,1} \ldots s_{i,T}$, where T is the length of state sequence, and 2) an observation y_i, sampled from the probability distribution $P(Y \mid x_i, M_{\theta_t})$. For each $x_i \in \mathcal{X}_N$, let $R(x_i) = \sum_{t=1}^{T} \gamma^t R(s_{i,t}, a_{i,t})$ be the total discounted rewards accumulated with the run x_i.

First, we compute the gradient of the value function with respect to the policy parameter θ,

$$\nabla_\theta V(s_0, \theta) = \nabla_\theta \sum_{x \in \mathcal{X}} P_\theta(x) R(x) \tag{7}$$

$$= \sum_{x \in \mathcal{X}} \nabla_\theta \left(P_{\theta_t}(x) \frac{P_\theta(x)}{P_{\theta_t}(x)} \right) R(x) \tag{8}$$

$$= \sum_{x \in \mathcal{X}} P_{\theta_t}(x) \frac{\nabla_\theta P_\theta(x)}{P_{\theta_t}(x)} R(x) \tag{9}$$

$$= \sum_{x \in \mathcal{X}} P_{\theta_t}(x) \frac{P_\theta(x)}{P_{\theta_t}(x)} \nabla_\theta \log(P_\theta(x)) R(x) \tag{10}$$

$$\approx \frac{1}{N} \sum_{x_i \in \mathcal{X}_N} \frac{P_\theta(x_i)}{P_{\theta_t}(x_i)} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi(a_{i,t} | s_{i,t}) \right) R(x_i), \tag{11}$$

where $\mathcal{X}_N = \{x_i, i = 1, \ldots, N\} \subseteq \mathcal{X}$ be a sample of N finite-length runs. Because the samples are obtained with policy π_{θ_t}, we use importance weighting in the second step. In the last step, we approximate the gradient using the N sampled trajectories.

The third term is the KL-divergence[5]. Taking derivate with respect to θ, we have:

$$\nabla_\theta D_{KL}(P_{\theta_t} || P_\theta) = \sum_{x \in \mathcal{X}} \left(\nabla_\theta P_{\theta_t}(x) \log \frac{P_{\theta_t}(x)}{P_\theta(x)} \right) \tag{12}$$

$$= - \sum_{x \in \mathcal{X}} P_{\theta_t}(x) \nabla_\theta \log P_\theta(x) \tag{13}$$

$$= - \mathbf{E}_{x \sim P_{\theta_t}} \left[\nabla_\theta \log(P_\theta(x)) \right] \tag{14}$$

$$\approx - \frac{1}{N} \sum_{x_i \in \mathcal{X}_N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_{i,t} | s_{i,t}) \right). \tag{15}$$

For the first term, we need to compute the derivative of the constraint with respect to the policy parameter.

$$\nabla_\theta \left(\alpha - \mathbf{E}_{y \sim M_\theta} \left[\mathbf{1}(\log \frac{P(y; M_\theta)}{P(y; M_0)} - \epsilon > 0) \right] \right) \tag{16}$$

$$= - \nabla_\theta \sum_y P(y; M_\theta) \left[\mathbf{1}(\log \frac{P(y; M_\theta)}{P(y; M_0)} - \epsilon > 0) \right] \tag{17}$$

$$= - \sum_y \nabla_\theta P(y; M_\theta) \left[\mathbf{1}(\log \frac{P(y; M_\theta)}{P(y; M_0)} - \epsilon > 0) \right] \tag{18}$$

$$= - \sum_{y \in U} \nabla_\theta P(y; M_\theta) \tag{19}$$

$$= - \sum_{y \in U} \nabla_\theta \sum_{x \in \mathcal{X}} P(y|x) P_\theta(x) \tag{20}$$

$$= - \sum_{y \in U} \sum_{x \in \mathcal{X}} P(y|x) \nabla_\theta P_\theta(x), \tag{21}$$

where $U = \{y \in \mathcal{Y} \mid \log \frac{P(y;M_\theta)}{P(y;M_0)} - \epsilon > 0\}$ is a set of observation sequences at which the detection condition is met.

Using the logarithm trick again, we have

$$\sum_{y\in U}\sum_{x\in\mathcal{X}} P(y|x)\nabla_\theta P_\theta(x) \tag{22}$$

$$= \sum_{y\in U}\sum_{x\in\mathcal{X}} P(y|x)P_\theta(x)\nabla_\theta \log(P_\theta(x)) \tag{23}$$

$$= \sum_{y\in U}\sum_{x\in\mathcal{X}} P_{\theta_t}(x)\frac{P_\theta(x)}{P_{\theta_t}(x)}P(y|x)\nabla_\theta \log P_\theta(x) \tag{24}$$

$$\approx \frac{1}{N}\sum_{x_i\in\mathcal{X}_N}\sum_{y\in U}\frac{P_\theta(x_i)}{P_{\theta_t}(x_i)}P(y|x_i)\nabla_\theta \log P_\theta(x_i). \tag{25}$$

The approximation requires the computation of $P(y|x_i)$, which is the probability of observation y given the run x_i, for all $y \in U$. In practice, the set U can be large and difficult to construct. We discuss two approaches to compute the approximation.

One approach is to uniformly sample a subset of U, called U_K, and compute the gradient as

$$(23) \approx \frac{1}{N}\sum_{x_i\in\mathcal{X}_N}\sum_{y_k\in U_K}\frac{P_\theta(x_i)}{P_{\theta_t}(x_i)}P(y_k|x_i)\nabla_\theta \log P_\theta(x_i)$$

$$= \frac{1}{N}\sum_{x_i\in\mathcal{X}_N}\sum_{y_k\in U_K}\frac{P_\theta(x_i)}{P_{\theta_t}(x_i)}P(y_k|x_i)\nabla_\theta(\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})).$$

Alternatively, it is noted that $P(y|x)P_{\theta_t}(x)$ is the joint probability $P_{\theta_t}(x, y)$.

$$(24) = \sum_{x\in\mathcal{X}}\sum_{y\in U} P_{\theta_t}(y,x)\cdot\frac{P_\theta(x)}{P_{\theta_t}(x)}\cdot\nabla_\theta \log P_\theta(x)$$

$$\approx \frac{1}{N}\sum_{x_i\in\mathcal{X}_N}\mathbf{1}(y_i\in U)\frac{P_\theta(x_i)}{P_{\theta_t}(x_i)}\cdot\nabla_\theta \log P_\theta(x_i)$$

$$= \frac{1}{N}\sum_{x_i\in\mathcal{X}_N}\mathbf{1}(y_i\in U)\frac{P_\theta(x_i)}{P_{\theta_t}(x_i)}\cdot\left(\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\right).$$

That is, for each sampled trajectory-observation pair (x_i, y_i) from the distribution P_{θ_t}, we check if $\log \frac{P_\theta(y)}{P_0(y)} - \epsilon > 0$, that is, using the distribution P_θ to determine if $y_i \in U$.

Finally, the gradient of the dual variable λ is calculated as the following:

$$\nabla_\lambda L^\beta(\theta,\lambda) \tag{26}$$

$$= \alpha - \Pr\left(\log\frac{P(Y;M_\theta)}{P(Y;M_0)} - \epsilon > 0, M_\theta\right) \tag{27}$$

$$= \alpha - \mathbf{1}\left(\log\frac{P(Y;M_\theta)}{P(Y;M_0)} > \epsilon\right) \tag{28}$$

$$\approx \alpha - \frac{1}{N}\sum_{y\in\overline{U}}\mathbf{1}\left(\log\frac{P(y;M^\theta)}{P(y;M_0)} > \epsilon\right), \tag{29}$$

where $\overline{U} = \{y_i \mid \log \frac{P(y_i;M_\theta)}{P(y_i;M_0)} > \epsilon\}$ is a set of *sampled* observations at which the detection condition is met.

Algorithm 1 summarizes the proposed Primal-Dual Covert Policy Gradient (Covert PG) method using two-time-scale updates. Two-time-scale updates are a common technique used by first-order algorithms for solving minimax or maximin problems (see, e.g., [7, 12, 17, 19, 26]). The inner-loop updates are carried out more frequently in order to compute an approximately optimal solution to the inner-loop optimization problem, which can be subsequently used for computing the gradient for the outer-loop updates. The convergence analysis of the proposed primal-dual proximal policy gradient method is left as future work. While several convergence analyses of two-time-scale first-order methods exist, they do not directly apply to the problem studied in this paper due to different assumptions on the optimization problem such as strong convexity/concavity of the inner problem [17] or that the primal-dual solution forms a strict local Nash equilibrium [7, 12].

In our numerical experiments, we chose to terminate the algorithm once the value of the Lagrangian did not change significantly between two consecutive iterations, which is determined by the choice of δ_0. λ_0 is the initial value of the dual variable.θ_0 is the initial policy used by the covert planner. The parameter d is the KL divergence target distance, and β is the coefficient of the KL divergence. If the KL divergence is significantly different from the target distance, β quickly adjusts. η_1 and η_2 is the learning rate of θ and λ. Finally, $(x)^+ = 0$ if $x \le 0$, $(x)^+ = x$ if $x > 0$.

Algorithm 1 Primal-dual proximal policy gradient for covert optimal planning

procedure PRIMAL-DUAL PROXIMAL POLICY GRADIENT($\lambda_0, \theta_0, \delta_0, d, \beta, \eta_1, \eta_2$)
  $t \leftarrow 1$
  $L^\beta(\theta_0, \lambda_0) \leftarrow \infty$
  $\delta \leftarrow \infty$
  while $|\delta| \ge \delta_0$ **do**
    for batches $b = 1, 2, ...m$ **do**
      Generate trajectories $\mathcal{X}_b^t$ based on θ_t
      Calculate gradient term $\nabla_\theta L^\beta(\theta, \lambda_t)$ given generated trajectories $\mathcal{X}_b$
      $\theta \leftarrow \theta + \eta_1 \nabla_\theta L^\beta(\theta, \lambda_t)$
    end for
    Calculate $L^\beta(\theta, \lambda_t)$
    Calculate $\nabla_\lambda L^\beta(\theta, \lambda_t)$ based on all trajectories $\cup_b \mathcal{X}_b^t$
    $\lambda_{t+1} \leftarrow (\lambda_t - \eta_2 \nabla_\lambda L^\beta(\theta, \lambda_t))^+$
    Calculate $D_{KL}(P_{\theta_t}||P_\theta)$
    if $D_{KL}(P_{\theta_t}||P_\theta) \le d/1.5$ **then**
      $\beta \leftarrow \beta/2$
    end if
    if $D_{KL}(P_{\theta_t}||P_\theta) \ge d \times 1.5$ **then**
      $\beta \leftarrow \beta \times 2$
    end if
    $\theta_{t+1} \leftarrow \theta$
    $\delta \leftarrow |L^\beta(\theta, \lambda_t) - L^\beta(\theta, \lambda_{t-1})|$
    $t \leftarrow t + 1$
  end while
end procedure

4 EXPERIMENT VALIDATION

We demonstrate our solutions using a gridworld planning environment, depicted in Figure 2. A state of the agent is denoted by (row, col). The adversary can move in one of four compass directions. When given the action "N", the adversary enters the intended cell with a $1-2\kappa$ probability and enters the neighboring cells, which are the west and east cells, with probability κ. Different values of stochasticity parameter κ are used in the experiment. The purple areas are bouncing walls, which the agents can not enter. If the agent takes an action that crashes into the bouncing walls or boundaries, the agent will stay in his previous state. The black cells are penalty states, in which the agent receives a penalty if he enters the state. The robot icon in the figure indicates the agent's initial state (for both the nominal user and the adversary).

For both the adversary and the normal user, the cost of each action is 0.2. They incur a cost of 2 if they enter the black states and receive a reward of 20 upon reaching their respective goals. The discounting factor is 0.95. The normal user's goal is $(7, 0)$ (marked by the star icon) and the adversary's goal is located at $(0, 0)$ (marked by the red circle). The normal user's policy is computed as the entropy-regulated optimal policy to maximize the total discounted reward with respect to the normal user's reward function.

We consider the behavior of the agent is observed using a set of sensors, whose configuration and the covering areas are shown in Figure 3. The sensor icon indicates the sensor's location and the yellow area depicts the sensor's coverage. Each sensor possesses a default detection probability and can solely detect agents presented within its coverage. The detection probability decreases with the distance of the agent's location to the sensor's location. Once a sensor detects an agent, the defender obtains an observation value of 1 from that particular sensor. Consequently, our experimental setup results in the defender's observation taking the form (o_1, o_2, o_3), with each o_i corresponding to the observation from sensor i.

The detection probability for a given state is influenced by three key factors: the range of the sensor, the sensor's default detection probability (0.8 for all sensors), the distance between the current state and the sensor's location, and the surrounding environment (green states). For every unit of distance increase, the detection probability decreases by 0.05. A dark (resp. light) green state further decreases the probability by $d_1 = 0.2$ (resp. $d_2 = 0.1$). For example, assume sensors one, two, and three are located at $(4, 0)$, $(6, 4)$, $(1, 4)$ respectively, if an agent is at state $(6, 3)$, which is not in the range of sensor one and three, thus the observations received from sensor one and three are 0. For sensor 2, the probability of detecting the agent is $0.8 - 0.05 \times 1 - 0.2 = 0.55$.

With this stochastic environment and sensor setup, we compute the optimal $(1-\alpha)$-covert policy given $\alpha = 0.2$ and different system stochasticity parameters $\kappa = 0.05, 0.1, 0.15$. In the experiments, the learning rate of θ is 0.005, and the learning rate of λ is 0.01. In each iteration, 800 trajectories are generated. These trajectories are divided into 20 batches to update θ in the primal-dual proximal policy gradient computation (Line 5-9 in Algorithm 1).

First, we compute the optimal policies without the covert constraint and evaluate the probability of detection of these policies. Given a system's stochasticity parameter $\kappa = 0.05$, without the covert constraint, the adversary's deterministic optimal policy would

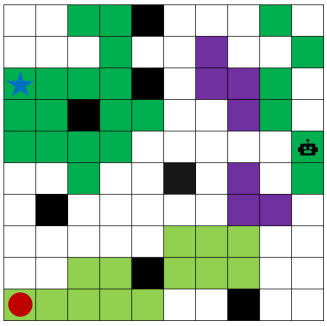

Figure 2: The 10×10 stochastic gridworld.

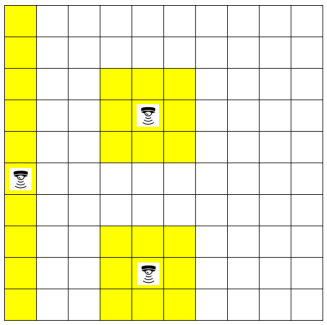

Figure 3: The Sensor Configuration.

attain a value of 6.76 at a cost of being detected with a probability close to 99%. Similarly, the adversary's value is 5.6 and 4.2 given $\kappa = 0.1$, and $\kappa = 0.15$ respectively if the adversary takes deterministic optimal policies in these two environments. The detection probabilities are close to 99% under both cases. If the adversary uses the softmax optimal policy, then the values are close to the deterministic optimal policies but the detection probabilities are approximately 73%, given $\kappa = 0.05, 0.1$, and 0.15.

Then, we solve for 0.8-covert policy that ensures the probability of detection is no greater than $\alpha = 20\%$. We set the detection threshold κ to be 3, following from χ^2 distribution with 1 degree-of-freedom and confidence interval 92%. It is conservatively selected to model a detector that tolerates a relatively large false positive rate. The performance of the algorithm is empirically analyzed from three values: Lagrangian function value, adversary's expected value, and detection probability, as shown in Figures 4, 5, and 6. We select the entropy-regulated optimal policy as the initial policy. Thus, the values of policies are highest upon initialization. As the policy is updated over iterations to enforce the covertness constraint, the values decrease. The algorithm terminates in all three cases.

Figure 4 shows the Lagrangian value change over iterations given $\kappa = 0.05, 0.1$and 0.15. When $\kappa = 0.1$, and 0.15, the initial value of λ is 10. The values of the Lagrangian functions change initially increases and then decreases after 200 iterations. This is because the detection constraint is satisfied after 200 iterations and after that the value of λ decreases to 0, resulting in a decrease in the Lagrangian value. It is observed that the initial value of Lagrangian for $\kappa = 0.05$ is smallest, close to -14. This is because we have picked the initial value of λ to be 40 for $\kappa = 0.05$ to achieve a faster termination.

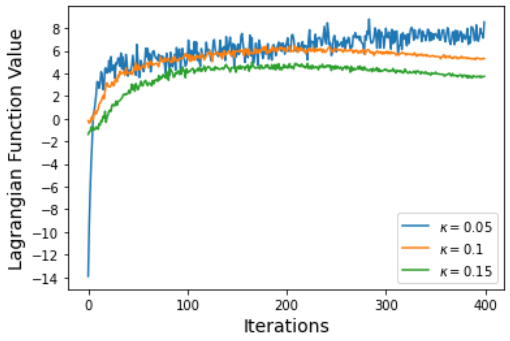

Figure 4: Lagrangian value over iterations given $\kappa = 0.05, 0.1, 0.15$.

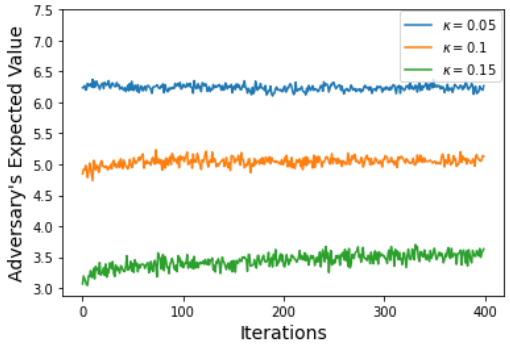

Figure 5: Expected value over iterations given $\kappa = 0.05, 0.1, 0.15$.

From Figure 5, we conclude the adversary's expected values are 6.3, 5.1, and 3.47, given $\kappa = 0.05$, 0.1, and 0.15 respectively. Compare the adversary's covert policies' expected values to the adversary's optimal expected values without enforcing covertness constraint (6.76, 5.6, and 4.2), the decreases are 7.5%, 9%, and 18%, for $\kappa = 0.05, 0.1$, and 0.15 respectively. This shows that in this current environment setting, the adversary can satisfy the detection constraint without too much loss in the task performance.

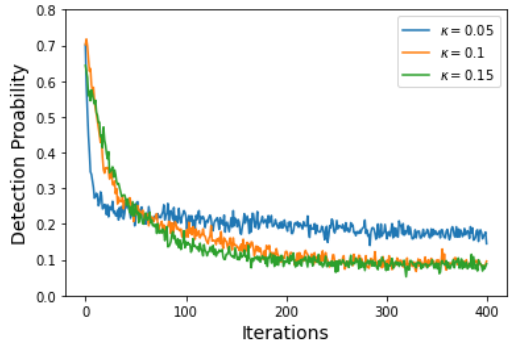

Figure 6: Detection probability over iterations given $\kappa = 0.05, 0.1, 0.15$.

Figure 6 depicts the change of the detection probabilities over iterations. The result shows all three initial policies have large detection probabilities, and the detection probabilities decrease quickly in the first 100 iterations and eventually satisfy the detection constraint. The final detection probabilities are 0.168, 0.089, and 0.088, given $\kappa = 0.05, 0.1$, and 0.15 respectively. The policies obtained upon termination all satisfy the 0.8-covertness.

To test the sensitivity of the policies concerning different system stochasticities, we evaluated the detection probabilities of the computed optimal policies for $\kappa = 0.05, 0.1$ and 0.15, under different levels of system stochasticity. The results are presented in Table 1. We observe that if the policy is optimized for the system with a stochasticity level κ, then this policy can still ensure covertness for $\kappa' \geq \kappa$ (as shown in Boldface in the table). We hypothesize that the increased noise can aid the covertness of the policy. However, verifying this hypothesis requires further analysis and more general classes of MDPs other than the gridworld dynamics. We leave this to future investigation.

The experiments are conducted using Python on a Windows 10 machine with Intel(R) Core (TM) i7-11700K CPU and 32 GB RAM. The average running time for 400 iterations is approximately 24 hours. The running time varies slightly with different system stochasticities due to differences in trajectory length.

κ	0.05	0.1	0.15
π^* given $\kappa = 0.05$	**0.168 ± 0.011**	**0.077 ± 0.006**	**0.083 ± 0.007**
π^* given $\kappa = 0.1$	0.289 ± 0.018	**0.089 ± 0.012**	**0.087 ± 0.009**
π^* given $\kappa = 0.15$	0.332 ± 0.011	0.093 ± 0.009	**0.088 ± 0.008**

Table 1: The detection probabilities obtained by evaluating different policies in different stochastic dynamics for the system.

5 CONCLUSION

We study a class of covert planning problems against imperfect observers and develop a covert primal-dual gradient method that optimizes task performance given certain covertness constraints. Despite the proof that finite-memory policies can be more powerful than memoryless policies under the covertness constraint, our solution is limited to finding an optimal and covert Markov policy. Future work may investigate approaches to efficiently search finite-memory policy space under covert constraints. By analyzing the covert policy, observer can consider whether it is possible to use the covert policy as a counterexample to improve the imperfect observer and eliminate such covert policies. Additionally, through empirical analysis, one can investigate theoretically how the detection probability can be influenced by the stochasticity in the system and different levels of noise in the observations.

ACKNOWLEDGEMENT

Research was sponsored by the Army Research Office under Grant Number W911NF-22-1-0034 and the Army Research Laboratory under Cooperative Agreement Number W911NF-22-2-0233. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] Wael Al Enezi and Clark Verbrugge. 2022. Stealthy path planning against dynamic observers. In *Proceedings of the 15th ACM SIGGRAPH Conference on Motion, Interaction and Games*. ACM, Guanajuato Mexico, 1–9. https://doi.org/10.1145/3561975.3562948

[2] Mohamed Al Marzouqi and Ray A Jarvis. 2011. Robotic covert path planning: A survey. In *2011 IEEE 5th international conference on robotics, automation and mechatronics (RAM)*. IEEE, 77–82.

[3] Dimitri P Bertsekas. 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press.

[4] Shaunak D Bopardikar, Francesco Bullo, and Joao P Hespanha. 2008. On discrete-time pursuit-evasion games with sensing limitations. *IEEE Transactions on Robotics* 24, 6 (2008), 1429–1439.

[5] Imre Csiszár. 1975. I-divergence geometry of probability distributions and minimization problems. *The annals of probability* (1975), 146–158.

[6] Dmitriy Drusvyatskiy and Lin Xiao. 2023. Stochastic optimization with decision-dependent distributions. *Mathematics of Operations Research* 48, 2 (2023), 954–998.

[7] Tanner Fiez and Lillian J Ratliff. 2021. Local Convergence Analysis of Gradient Descent Ascent with Finite Timescale Separation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=AWOSz_mMAPx

[8] Cheng-Der Fuh. 2003. SPRT and CUSUM in Hidden Markov Models. *The Annals of Statistics* 31, 3 (2003), 942–977. https://www.jstor.org/stable/3448426 Publisher: Institute of Mathematical Statistics.

[9] Tobias Gabi Goobar and Samuel Söderberg. 2021. Knowledge Based Strategies in Grid-Based Pursuit-Evasion Games of Imperfect Information. , 609-622 pages.

[10] Brian P Gerkey, Sebastian Thrun, and Geoff Gordon. 2006. Visibility-based pursuit-evasion with limited field of view. *The International Journal of Robotics Research* 25, 4 (2006), 299–315.

[11] Volkan Isler, Sampath Kannan, and Sanjeev Khanna. 2006. Randomized Pursuit-Evasion with Local Visibility. *SIAM Journal on Discrete Mathematics* 20, 1 (Jan. 2006), 26–41. https://doi.org/10.1137/S0895480104442169

[12] Chi Jin, Praneeth Netrapalli, and Michael Jordan. 2020. What Is Local Optimality in Nonconvex-Nonconcave Minimax Optimization?. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4880–4889.

[13] Mustafa O. Karabag, Melkior Ornik, and Ufuk Topcu. 2021. Deception in Supervisory Control. *IEEE Trans. Automat. Control* (2021), 1–1. https://doi.org/10.1109/TAC.2021.3057991 Conference Name: IEEE Transactions on Automatic Control.

[14] Mustafa O. Karabag, Melkior Ornik, and Ufuk Topcu. 2023. Exploiting Partial Observability for Optimal Deception. *IEEE Trans. Automat. Control* 68, 7 (July 2023), 4443–4450. https://doi.org/10.1109/TAC.2022.3209959 Conference Name: IEEE Transactions on Automatic Control.

[15] Mhr. Khouzani and Pasquale Malacaria. 2017. Leakage-Minimal Design: Universality, Limitations, and Applications. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, Santa Barbara, CA, 305–317. https://doi.org/10.1109/CSF.2017.40

[16] Alan Lewis and Tim Miller. 2023. Deceptive Reinforcement Learning in Model-Free Domains. *arXiv preprint arXiv:2303.10838* (2023).

[17] Tianyi Lin, Chi Jin, and Michael I. Jordan. 2020. Near-Optimal Algorithms for Minimax Optimization. In *Proceedings of Thirty Third Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 125)*. PMLR, 2738–2779.

[18] Zhengshang Liu, Yue Yang, Tim Miller, and Peta Masters. 2021. Deceptive reinforcement learning for privacy-preserving planning. *arXiv preprint arXiv:2102.03022* (2021).

[19] Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. 2019. Solving a Class of Non-Convex Min-Max Games Using Iterative First Order Methods. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[20] Miquel Ramırez and Hector Geffner. [n.d.]. Goal Recognition over POMDPs: Inferring the Intention of a POMDP Agent. ([n. d.]).

[21] Yagiz Savas, Michael Hibbard, Bo Wu, Takashi Tanaka, and Ufuk Topcu. 2022. Entropy Maximization for Partially Observable Markov Decision Processes. *IEEE Trans. Automat. Control* 67, 12 (Dec. 2022), 6948–6955. https://doi.org/10.1109/TAC.2022.3183564 Conference Name: IEEE Transactions on Automatic Control.

[22] Yagiz Savas, Mustafa O Karabag, Brian M Sadler, and Ufuk Topcu. 2022. Deceptive Planning for Resource Allocation. *arXiv preprint arXiv:2206.01306* (2022).

[23] Yagiz Savas, Christos K Verginis, and Ufuk Topcu. 2022. Deceptive decision-making under uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 5332–5340.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[25] Gita Sukthankar, Christopher Geib, Hung Hai Bui, David Pynadath, and Robert P Goldman. 2014. *Plan, activity, and intent recognition: Theory and practice*. Newnes.

[26] Kiran K Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. 2019. Efficient Algorithms for Smooth Minimax Optimization. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 12680–12691.

[27] Qingsi Wang and Mingyan Liu. 2014. Learning in hide-and-seek. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. 217–225. https://doi.org/10.1109/INFOCOM.2014.6847942 ISSN: 0743-166X.