Non-Cooperative Inverse Reinforcement Learning

Xiangyuan Zhang **Kaiqing Zhang** **Erik Miehling** **Tamer Başar**

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
{xz7,kzhang66,miehling,basar1}@illinois.edu

Abstract

Making decisions in the presence of a strategic opponent requires one to take into account the opponent's ability to actively mask its intended objective. To describe such strategic situations, we introduce the non-cooperative inverse reinforcement learning (N-CIRL) formalism. The N-CIRL formalism consists of two agents with completely misaligned objectives, where only one of the agents knows the true objective function. Formally, we model the N-CIRL formalism as a zero-sum Markov game with one-sided incomplete information. Through interacting with the more informed player, the less informed player attempts to both infer and optimize the true objective function. As a result of the one-sided incomplete information, the multi-stage game can be decomposed into a sequence of single-stage games expressed by a recursive formula. Solving this recursive formula yields the value of the N-CIRL game and the more informed player's equilibrium strategy. Another recursive formula, constructed by forming an auxiliary game, termed the dual game, yields the less informed player's strategy. Building upon these two recursive formulas, we develop a computationally tractable algorithm to approximately solve for the equilibrium strategies. Finally, we demonstrate the benefits of our N-CIRL formalism over the existing multi-agent IRL formalism via extensive numerical simulation in a novel cyber security setting.

1 Introduction

In any decision-making problem, the decision-maker's goal is characterized by some underlying, potentially unknown, objective function. In machine learning, ensuring that the learning agent does what the human intends it to do requires specification of a *correct* objective function, a problem known as *value alignment* [42, 37]. Solving this problem is nontrivial even in the simplest (single-agent) reinforcement learning (RL) settings [3] and becomes even more challenging in multi-agent environments [19, 43, 45]. Failing to specify a correct objective function can lead to unexpected and potentially dangerous behavior [38].

Instead of specifying the correct objective function, recent research has taken an alternative perspective of letting the agent learn the intended task from observed behavior of other agents and/or human experts, a concept known as *inverse reinforcement learning* (IRL) [31]. IRL describes a setting where one agent is attempting to learn the true reward function by observing trajectories of sample behavior of another agent, termed *demonstrations*, under the assumption that the observed agent is acting (optimally) according to the true reward function. More recently, the concept of *cooperative inverse reinforcement learning* (CIRL) was introduced in [9]. Instead of passively learning from demonstrations (as is the case in IRL), CIRL allows for agents to *interact* during the learning process, which results in improved performance over IRL. CIRL inherently assumes that the two agents are on the same team, that is, the expert is actively trying to help the agent learn and achieve some common goal. However, in many practical multi-agent decision-making problems, the objec-

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

tives of the agents may be misaligned, and in some settings completely opposed [22, 21, 40, 44] (*e.g.*, zero-sum interactions in cyber security [2, 27]). In such settings, the expert is still trying to achieve its objective, but may act in a way so as to make the agent think it has a different objective. Development of a non-cooperative analogue of CIRL to describe learning in these settings has not yet been investigated.

In this paper, we introduce the *non-cooperative inverse reinforcement learning* (N-CIRL) formalism. The N-CIRL formalism consists of two agents with completely misaligned objectives, where only one agent knows the true reward function. The problem is modeled as a zero-sum Markov game with one-sided incomplete information. In particular, at any stage of the game the information available to the player that does not know the true reward function, termed the *less informed* player, is contained within the information available to the player that knows the reward function, termed the *more informed* player. This one-sided information structure allows for a simplification in the form of the beliefs (compared to the fully general asymmetric information case [10]) and, more importantly, allows one to define strategies as the stage-wise solutions to two recursive equations [7, 35]. By taking advantage of the structure of these recursive equations, and their corresponding solutions (fixed points), computationally tractable algorithms for approximately computing the players' strategies are developed.

Our primary motivation for developing the N-CIRL formalism is cyber security. In particular, we are interested in settings where the attacker has some intent that is unknown to the defender. In reality, the motivation of attackers can vary significantly. For example, if the attacker is financially motivated, its goal may be to find personal payment details, whereas if the attacker is aiming to disrupt the normal operation of a system, reaching a computer responsible for controlling physical components may be of more interest. This variability in what the attacker values is captured by the N-CIRL formalism through an *intent* parameter that serves to parameterize the attacker's true reward function. The defender, who does not know the true intent, then faces the problem of learning the attacker's intent while simultaneously defending the network. The attacker, knowing this, aims to reach its goal but may behave in a way that makes its true intent as unclear as possible.[1]

Throughout the remainder of the paper, we will refer to the more informed player as the *attacker* and the less informed player as the *defender*. While we use the cyber security example throughout the paper, this is primarily for ease of exposition. The results presented here apply to any zero-sum Markov game setting where one player does not know the true reward.

Limitations of Existing IRL Approaches. The application of N-CIRL to cyber security is especially fitting due to the challenges associated with collecting useful attack data. Obtaining accurate attack logs is a computationally formidable task [16]. Furthermore, learning from sample equilibrium behavior, as is done in the *multi-agent inverse reinforcement learning* (MA-IRL) settings of [21, 41, 20], is only useful if the goal(s) do not change between learning and execution/deployment. Such an assumption is not appropriate in cyber security settings – the attacker's goal as well as the overall system structure, may frequently change. This non-stationary behavior necessitates the ability to intertwine learning and execution. N-CIRL provides a formalism for specifying actions that adapt to the information revealed during the game. We illustrate this adaptivity via an illustrative example in Sec. A and extensive numerical results in Sec. 5.

Contribution. The contribution of the present work is three-fold: **1)** We propose a new formalism for IRL, termed N-CIRL, that describes how two competing agents make strategic decisions when only one of the agents possesses knowledge of the true reward; **2)** By recognizing that N-CIRL is a zero-sum Markov game with one-sided incomplete information, we leverage the recursive structure of the game to develop a computationally tractable algorithm, termed *non-cooperative point-based value iteration* (NC-PBVI), for computing both players' strategies; **3)** We demonstrate in a novel cyber security model that the adaptive strategies obtained from N-CIRL outperform strategies obtained from existing multi-agent IRL techniques.

2 Related Work

Decision-making when the agents are uncertain of the true objective function(s) has been extensively studied within the fields of both RL and game theory. One standard and popular way to infer the

[1] An interesting real-world example of such behavior was Operation Fortitude in World War II [12].

actual reward function is via inverse RL, the idea of which was first introduced by [17] under the title of inverse optimal control. Later, [31] formally introduced the notion of IRL with the goal of inferring the reward function being optimized by observing the behavior of an actor, termed an *expert*, over time [31, 1, 33, 8]. Fundamental to the IRL setting is the assumption that the agent inferring the reward *passively* observes the expert's behavior, while the expert behaves optimally in its own interest without knowing that the agent will later use the observed behavior to learn.

As pointed out in [9], such an assumption is not valid in certain cooperative settings where the agent and the expert are able to interact in order to achieve some common objective. In fact, IRL-type solutions were shown to be suboptimal and generally less effective at instilling the knowledge of the expert to the agent [9]. As argued by [9], the value alignment problem with cooperative agents is more appropriately viewed as an interactive decision-making process. The proposed formalism, termed CIRL, is formulated as a two-player game of partial information with a common reward function[2]. Due to the special structure of CIRL, the problem can be transformed into a *partially observable Markov decision process* (POMDP), see [30, 9], allowing for single-agent RL algorithms to be applied. Further improvements in computational efficiency can be achieved by exploiting the fact that the expert expects the agent to respond optimally [23].

Inverse RL under a non-cooperative setting has not received as much attention as its cooperative counterpart. A recent collection of work on multi-agent IRL (MA-IRL) [21, 41, 20] addresses the problem of IRL in stochastic games with multiple (usually more than two) agents. Distinct from our N-CIRL setting, MA-IRL aims to recover the reward function of multiple agents under the assumption that the demonstrations are generated from the *Nash equilibrium* strategies. Moreover, in the MA-IRL formalism, agents determine their strategies based on the inferred reward function, *i.e.*, regarding the inferred reward as some fixed ground truth. In contrast, under our N-CIRL setting, only one agent is unaware of the true reward function. Furthermore, the goal of the less informed player in N-CIRL goes beyond just *inferring* the true reward function; its ultimate goal is to determine an optimal strategy against a worst-case opponent who possesses a private reward.

From a game theoretic perspective, the N-CIRL formalism can be viewed as a stochastic dynamic game with asymmetric information, see [6, 28, 29, 4] and references therein. In particular, N-CIRL lies within the class of games with one-sided incomplete information [35, 39, 34, 14, 13]. This type of game allows for a simplified belief and allows the game to be decomposed into a sequence of single-stage games [35, 13]. In particular, our N-CIRL formalism can be recognized as one of the game settings discussed in [35], in which a *dual game* was formulated to solve for the less informed player's strategy. Our algorithm for computing the defender's strategy is built upon this formulation, and can be viewed as one way to approximately solve the dual game.

3 Non-Cooperative Inverse Reinforcement Learning

In this section, we introduce the N-CIRL formalism and describe its information structure. As will be shown, the information structure of N-CIRL admits compact information states for each player.

3.1 N-CIRL Formulation

The N-CIRL formalism is modeled as a two-player zero-sum Markov game with one-sided incomplete information. In particular, the attacker knows the true reward function, while the defender does not. In the context of the cyber security setting of this paper, the reward function is assumed to be parameterized by an intent parameter that is only known to the attacker. The N-CIRL formalism is described by the tuple $\langle \mathcal{S}, \{\mathcal{A}, \mathcal{D}\}, \mathcal{T}(\cdot \mid \cdot, \cdot, \cdot), \{\Theta, R(\cdot, \cdot, \cdot, \cdot; \cdot)\}, \mathcal{P}_0(\cdot, \cdot), \gamma \rangle$, where

- $\mathcal{S}$ is the finite set of states; $s \in \mathcal{S}$.
- $\mathcal{A}$ is the finite set of actions for the attacker A; $a \in \mathcal{A}$.
- $\mathcal{D}$ is the finite set of actions for the defender D; $d \in \mathcal{D}$.
- $\mathcal{T}(s' \mid s, a, d)$ is the conditional distribution of the next state s' given current state s and actions a, d.
- Θ is the finite set of intent parameters that parameterize the reward function; the true intent parameter $\theta \in \Theta$ is only observed by the attacker.

[2] Also known as a *team* problem [24].

- $R(s, a, d, s'; \theta)$ is the parameterized reward function that maps the current state $s \in \mathcal{S}$, actions $(a, d) \in \mathcal{A} \times \mathcal{D}$, next state $s' \in \mathcal{S}$, and parameter $\theta \in \Theta$ to a reward for the attacker.
- $\mathcal{P}_0$ is the distribution over the initial state s_0 and the true reward parameter θ, assumed to be common knowledge between A and D.
- $\gamma \in [0, 1)$ is the discount factor.

The game proceeds as follows. Initially, a state-parameter pair (s_0, θ) is sampled from the prior distribution $\mathcal{P}_0$. The state s_0 is publicly observed by both players, whereas the intent parameter θ is only observed by the attacker.[3] For each stage, the attacker and the defender act simultaneously, choosing actions $a \in \mathcal{A}$ and $d \in \mathcal{D}$. Note that the action sets may be state-dependent, *i.e.*, $a \in \mathcal{A}(s)$, $d \in \mathcal{D}(s)$. Given both actions, the current state s transitions to a successor state s' according to the transition model $\mathcal{T}(s' \mid s, a, d)$. The attacker receives a bounded reward $R(s, a, d, s'; \theta)$; the defender receives the reward $-R(s, a, d, s'; \theta)$ (incurs a cost $R(s, a, d, s'; \theta)$). Neither player observes rewards during the game. Before each subsequent stage, both players are informed of the successor state s' and the actions from the previous stage. While both players are aware of the current state, only the attacker is aware of the true intent parameter $\theta \in \Theta$. This results in the defender possessing incomplete information, requiring it to maintain a belief over the true intent parameter. The goals of the attacker and the defender are to maximize and minimize the expected γ-discounted accumulated reward induced by R, respectively.

3.2 The Information Structure of N-CIRL

The N-CIRL formalism falls within the class of partially observable stochastic games [10]. In such games, perfect recall ensures that behavioral strategies, *i.e.*, strategies that mix over actions, are outcome equivalent to mixed strategies, *i.e.*, strategies that mix over pure strategies [18]. As a result, players in N-CIRL can restrict attention to behavioral strategies, defined for each stage t as $\pi_t^A : \mathcal{I}_t^A \to \Delta(\mathcal{A})$ and $\pi_t^D : \mathcal{I}_t^D \to \Delta(\mathcal{D})$, where $\mathcal{I}_t^A$ (resp. $\mathcal{I}_t^D$) represents the space of information available to the attacker (resp. defender) at stage t and $\Delta(\mathcal{A})$, $\Delta(\mathcal{D})$ represent distributions over actions. Given any realized information sequences $I_t^A \in \mathcal{I}_t^A$ and $I_t^D \in \mathcal{I}_t^D$, represented as

$$I_t^A = \big(s_0, \theta, a_0, d_0, \ldots, a_{t-1}, d_{t-1}, s_t\big), \quad I_t^D = \big(s_0, a_0, d_0, \ldots, a_{t-1}, d_{t-1}, s_t\big)$$

the defender's information is always contained within the attacker's information for any stage t, *i.e.*, there is one-sided incomplete information. Furthermore, note that the attacker has complete information (it knows everything that has happened in the game); its information at stage t is the full history of the game at t, denoted by $I_t = \big(s_0, \theta, a_0, d_0, \ldots, a_{t-1}, d_{t-1}, s_t\big) \in \mathcal{I}_t = \mathcal{I}_t^A$.

Information States. In general, games of incomplete information require players to reason over the entire belief hierarchy, that is, players' decisions not only depend on their beliefs on the state of nature, but also on the beliefs on others' beliefs on the state and nature, and so on [25, 10]. Fortunately, players do not need to resort to this infinite regress in games of one-sided incomplete information. Instead, each player is able to maintain a compact state of knowledge, termed an *information state*, that is sufficient for making optimal decisions. The more informed player maintains a pair consisting of the observable state and a distribution over the private state [35, 39]. The less informed player, through construction of a *dual game* (discussed in Sec. 4.2), maintains a pair consisting of the observable state and a vector (in Euclidean space) of size equal to the number of private states [7, 35]. In the context of N-CIRL, the attacker's information state at each stage is a pair, denoted by (s, b), in the space $\mathcal{S} \times \Delta(\Theta)$ whereas the defender's information state at each stage (of the dual game) is a pair, denoted by (s, ζ), in the space $\mathcal{S} \times \mathbb{R}^{|\Theta|}$.

4 Solving N-CIRL

The theoretical results used to solve the CIRL problem [30, 9] do not extend to the N-CIRL setting. As outlined in [9], the form of CIRL allows one to convert the problem into a centralized control problem [30].The problem can then be solved using existing techniques from reinforcement learning.

In N-CIRL, such a conversion to a centralized control problem is not possible; one is instead faced with a dynamic game. As we show in this section, the one-sided incomplete information allows

[3]The intent parameter θ is further assumed to be fixed throughout the problem.

one to recursively define both the value of the game and the attacker's strategy. One can further recursively define the defender's strategy via the construction and sequential decomposition of a dual game. The two recursive formulas permit the development of a computational procedure, based on linear programming, for approximating both players' strategies.

4.1 Sequential Decomposition

Solving a game involves finding strategies for all players, termed a *strategy profile*, such that the resulting interaction is in (Nash) equilibrium. A strategy profile for N-CIRL is defined as follows.

Definition 4.1 (Strategy Profile)**.** *A strategy profile, denoted by (σ^A, σ^D), is a pair of strategies $\sigma^A = (\pi_0^A, \pi_1^A, \ldots)$ and $\sigma^D = (\pi_0^D, \pi_t^D, \ldots)$, where π_t^A and π_t^D are behavioral strategies as defined in Sec. 3.2.*

A simplification of behavioral strategies are strategies that only depend on the most recent information rather than the entire history. These strategies, termed *one-stage strategies*, are defined below.

Definition 4.2 (One-Stage Strategies)**.** *The one-stage strategies of the attacker and defender are denoted by $\bar{\pi}^A : \mathcal{S} \times \Theta \to \Delta(\mathcal{A})$ and $\bar{\pi}^D : \mathcal{S} \to \Delta(\mathcal{D})$, respectively. The pair $(\bar{\pi}^A, \bar{\pi}^D)$ is termed a one-stage strategy profile.*

Due to the information structure of N-CIRL, the attacker's information state (s, b) can be updated using one-stage strategies instead of the full strategy profile. In fact, as illustrated by Lemma 1, the update of the attacker's information state only depends on the attacker's one-stage strategy $\bar{\pi}^A$, the attacker's action a, and the successor state s'.

Lemma 1 (Information State Update)**.** *Given the attacker's one-stage strategy profile $\bar{\pi}^A$, the current attacker's information state $(s, b) \in \mathcal{S} \times \Delta(\Theta)$, the attacker's action $a \in \mathcal{A}$, and the new state $s' \in \mathcal{S}$, the attacker's updated information state is $(s', b') \in \mathcal{S} \times \Delta(\Theta)$ where the posterior b' is computed via the function $\tau : \mathcal{S} \times \Delta(\Theta) \times \mathcal{A} \to \Delta(\Theta)$, defined elementwise as*

$$b'(\vartheta) = \tau_\vartheta(s, b, a) = \frac{\bar{\pi}^A(a \mid s, \vartheta) b(\vartheta)}{\sum_{\vartheta' \in \Theta} \bar{\pi}^A(a \mid s, \vartheta') b(\vartheta')}. \tag{1}$$

The attacker's information state (s, b) also leads to the following definition of the value function $v : \mathcal{S} \times \Delta(\Theta) \to \mathbb{R}$ of the game

$$v(s, b) = \max_{\sigma^A} \min_{\sigma^D} \; \mathbb{E}\left[\sum_{t \geq 0} \gamma^t R(s_t, a_t, d_t, s_{t+1}; \theta) \,\middle|\, s_0 = s, \theta \sim b(\cdot) \right],$$

which denotes the minimax accumulated reward if the initial state is $s_0 = s$, and the belief over Θ is b. Note that the value exists as all spaces in the game are finite and the discount factor lies in $[0, 1)$ [35]. The value function v can be computed recursively via a sequential decomposition. In fact, it is given by the fixed point of a value backup operator $[Gv](s, b)$, as illustrated in Proposition 1 below. To differentiate from the dual game to be introduced in Sec. 4.2, we refer to the original N-CIRL game as the *primal* game and $[Gv](s, b)$ as the *primal* backup operator.

Proposition 1 (Sequential Decomposition of Primal Game)**.** *The primal game can be sequentially decomposed into a sequence of single-stage games. Specifically, the primal value function v satisfies the following recursive formula*

$$v(s, b) = [Gv](s, b) = \max_{\bar{\pi}^A} \min_{\bar{\pi}^D} \left\{ g_{\bar{\pi}^A, \bar{\pi}^D}(s, b) + \gamma V_{\bar{\pi}^A, \bar{\pi}^D}(v; s, b) \right\} \tag{2}$$

where $[Gv](s, b)$ is referred to as the primal value backup operator*, and $g_{\bar{\pi}^A, \bar{\pi}^D}(s, b)$, $V_{\bar{\pi}^A, \bar{\pi}^D}(v; s, b)$ correspond to the instantaneous reward and the expected value of the continuation game, respectively, defined as*

$$g_{\bar{\pi}^A, \bar{\pi}^D}(s, b) = \sum_{a, d, s', \vartheta} b(\vartheta) \bar{\pi}^A(a \mid s, \vartheta) \bar{\pi}^D(d \mid s) \mathcal{T}(s' \mid s, a, d) R(s, a, d, s'; \vartheta) \tag{3}$$

$$V_{\bar{\pi}^A, \bar{\pi}^D}(v; s, b) = \sum_{a, d, s', \vartheta} b(\vartheta) \bar{\pi}^A(a \mid s, \vartheta) \bar{\pi}^D(d \mid s) \mathcal{T}(s' \mid s, a, d) v(s', b') \tag{4}$$

where b' represents the posterior distribution on Θ as computed by Eq. (1)*.*

For purposes of constructing an algorithm, we need to establish some properties of the backup operator defined in Proposition 1. The following lemma ensures that each application of the primal value backup yields a closer approximation of the value of the game.

Lemma 2 (Contraction of Primal Backup Operator)**.** *The primal value backup operator* $[Gv](s, b)$*, defined in Eq.* (2)*, is a contraction mapping. As a result, iterating the operator converges to the value of the primal game that solves the fixed point equation* (2)*.*

Though conceptually correct, iterating the backup operator $[Gv](s, b)$ exactly does not lead to a computationally tractable algorithm, as the belief b lies in a continuous space with an infinite cardinality. Thus, an approximate value iteration algorithm is required for solving the fixed point equation (2). We will address this computational challenge in Sec. 4.3.

Another challenge in solving N-CIRL is that the fixed point problem, given by Eq. (2), cannot be solved by the defender. In fact, as pointed out in [35, Sec. 1.2], if the defender is unaware of the attacker's strategy, it cannot form the posterior on Θ. The following section discusses the formulation of an auxiliary game to address this challenge.

4.2 The Defender's Strategy

As shown by [7, 35], the defender's equilibrium strategy can be determined by construction of a dual game, characterized by a tuple $\langle \mathcal{S}, \{\mathcal{A}, \mathcal{D}\}, \mathcal{T}(\cdot \mid \cdot, \cdot, \cdot), \{\Theta, R(\cdot, \cdot, \cdot, \cdot; \cdot)\}, \zeta_0, \mathcal{P}_0^{\mathcal{S}}(\cdot), \gamma \rangle$. Note that the sets $\mathcal{S}, \mathcal{A}, \mathcal{D}, \Theta$, the reward function $R(\cdot, \cdot, \cdot, \cdot; \cdot)$, the discount factor γ, and the state transition distribution $\mathcal{T}$ are identical to those in the primal game. The quantity $\zeta_0 \in \mathbb{R}^{|\Theta|}$ is the parameter of the dual game, $\mathcal{P}_0^{\mathcal{S}}(\cdot) \in \Delta(\mathcal{S})$ is the initial distribution of the state s_0, which is obtained by marginalizing $\mathcal{P}_0(\cdot, \cdot)$ over θ, *i.e.*, $\mathcal{P}_0^{\mathcal{S}}(s) = \sum_{\theta \in \Theta} \mathcal{P}_0(s, \theta)$. The dual game proceeds as follows: at the initial stage, s_0 is sampled from $\mathcal{P}_0^{\mathcal{S}}$ and revealed to both players, the attacker *chooses* some $\theta \in \Theta$; then the game is played identically as the primal one, namely, both players choose actions $a \in \mathcal{A}$ and $d \in \mathcal{D}$ simultaneously, and the state transitions from s to s' following $\mathcal{T}(\cdot \mid s, a, d)$. Both players are then informed of the chosen actions and the successor state s'. Furthermore, a reward of $R(s, a, d, s'; \theta) + \zeta_0(\theta)$ is received by the attacker (thus $-R(s, a, d, s'; \theta) - \zeta_0(\theta)$ is incurred by the defender). Note that the value of θ is *decided* and only known by the attacker, instead of being drawn from some probability distribution. This is one of the key differences from the primal game.

The value function of the dual game, denoted by $w : \mathcal{S} \times \mathbb{R}^{|\Theta|} \to \mathbb{R}$, is defined as the maximin γ-discounted accumulated reward received by the attacker, if the state starts from some $s_0 = s \in \mathcal{S}$ and the game parameter is $\zeta_0 = \zeta \in \mathbb{R}^{|\Theta|}$. The value $w(s, \zeta)$ exists since the dual game is finite [35]. Similarly as in Proposition 1, the dual game value function w also satisfies a recursive formula, as formally stated below.

Proposition 2 (Sequential Decomposition of Dual Game)**.** *The dual game can be decomposed into a sequence of single-stage games. Specifically, the dual value function* w *satisfies the following recursive formula*

$$w(s, \zeta) = [Hw](s, \zeta) = \min_{\bar{\pi}^D, \xi} \max_{\mu} \left\{ h_{\bar{\pi}^D, \mu}(s, \zeta) + \gamma W_{\bar{\pi}^D, \mu}(w, \xi; s) \right\} \tag{5}$$

where $[Hw](s, \zeta)$ *is referred to as the* dual value backup operator*,* $\bar{\pi}^D(\cdot|s) \in \Delta(\Theta)$*,* $\xi \in \mathbb{R}^{\mathcal{S} \times \mathcal{A} \times \Theta}$ *are decision variables with* $\xi_{a,s} \in \mathbb{R}^{|\Theta|}$ *the* (a, s)*th vector of* ξ*,* $\mu \in \Delta(\mathcal{A} \times \Theta)$*. Moreover,* $h_{\bar{\pi}^D, \mu}(s, \zeta)$ *and* $W_{\bar{\pi}^D, \mu}(w, \xi; s)$ *are defined as*

$$h_{\bar{\pi}^D, \mu}(s, \zeta) := \sum_{a, \vartheta} \mu(a, \vartheta) \Big(\zeta(\vartheta) + \sum_{d, s'} \bar{\pi}^D(d \mid s) \mathcal{T}(s' \mid s, a, d) R(s, a, d, s'; \vartheta) \Big), \tag{6}$$

$$W_{\bar{\pi}^D, \mu}(w, \xi; s) := \sum_{a, d, s', \vartheta} \mu(a, \vartheta) \bar{\pi}^D(d \mid s) \mathcal{T}(s' \mid s, a, d) \big(w(\xi_{a,s'}, s') - \xi_{a,s'}(\vartheta) \big). \tag{7}$$

The recursive formula above allows for a stage-wise calculation of the defender's strategy $\bar{\pi}^D$. In particular, by [35], the defender's equilibrium strategy obtained from the dual game is indeed its equilibrium strategy for the primal game. Moreover, the pair (s, ζ) is indeed the information state of the defender in the dual game. More importantly, the formula of Eq.(5) does not involve the update of the belief b, as opposed to Eq. (2). Instead, the vector ξ plays the similar role as the updated belief

b', which can be *calculated* by the defender as a decision variable. This way, as the defender plays the N-CIRL game, it can calculate its equilibrium strategy by only observing the attacker's *action* a and the successive state s', with no need to know the attacker's *strategy*. Besides, the defender's strategy $\bar{\pi}^D$, the decision variable $\mu \in \Delta(\mathcal{A} \times \Theta)$ in Eq. (5), is essentially the attacker's strategy in the dual game, *i.e.*, given ζ, the attacker has the flexibility to choose both $\theta \in \Theta$ and action $a \in \mathcal{A}$.

The remaining issue is to solve the dual game by solving the fixed point equation (5). As with the primal counterpart, the dual value backup operator is also a contraction mapping, as shown below.

Lemma 3 (Contraction of Dual Backup Operator)**.** *The value backup operator* $[Hw](s, \zeta)$*, defined in Eq.* (5)*, is a contraction mapping. As a result, iterating the operator converges to the dual game value that solves the fixed point equation* (5)*.*

By Lemma 3, the defender's strategy can be obtained by iterating the backup operator. However, as ζ and ξ both lie in continuous spaces, such an iterative approach is not computationally tractable. This motivates the approximate value iteration algorithm to be introduced next.

4.3 Computational Procedure

One standard algorithm for computing strategies in single-agent partially observable settings is the *point-based value iteration* (PBVI) algorithm [32]. The standard PBVI algorithm approximates the value function, which is convex in the beliefs, using a set of α-vectors. While the value functions v and w in N-CIRL also have desirable structures (as shown in [35], $v : \mathcal{S} \times \Delta(\Theta) \to \mathbb{R}$ is concave on $\Delta(\Theta)$ and $w : \mathcal{S} \times \mathbb{R}^{|\Theta|} \to \mathbb{R}$ is convex on $\mathbb{R}^{|\Theta|}$ for each $s \in \mathcal{S}$), the standard PBVI algorithm does not directly apply to N-CIRL. The challenge arises from the update of the α-vector set in PBVI, which requires carrying out one step of the Bellman update [32], for every action-observation pair of the agent. The corresponding Bellman update in N-CIRL is the primal backup operator of Eq. (2), which requires knowledge of the defender's strategy, not known by the attacker.

To address this challenge, we develop a modified version of PBVI, termed *non-cooperative PBVI* (NC-PBVI), in which the value functions v and w are each approximated by a set of *information state-value pairs*, *i.e.,* (s, b) and (s, ζ), instead of α-vectors. Importantly, updating the set of pairs only requires evaluations at individual information states, avoiding the need to know the opponent's strategies.

The evaluations can be approximated using linear programming. Specifically, to approximate the value function of the primal game, NC-PBVI updates the value at a given attacker's information state (s, b) by solving the primal backup operator of Eq. (2). Using a standard reformulation, the minimax operation in the primal backup operator can be approximately solved via a linear program, denoted by $P_A(s, b)$. Similarly, one can approximately solve the dual game's fixed point equation, Eq. (5), at a given defender's information state (s, ζ) via another linear program, denoted by $P_D(s, \zeta)$. For notational convenience, define $T_{sad}(s') = \mathcal{T}(s' \mid s, a, d)$, $P^{\vartheta}_{sad} = \sum_{s'} \mathcal{T}(s' \mid s, a, d) R(s, a, d, s'; \vartheta)$, $A_{s\vartheta}(a) = \bar{\pi}^A(a \mid s, \vartheta) b(\vartheta)$, and $D_s(d) = \bar{\pi}^D(d \mid s)$. The two linear programs, $P_A(s, b)$ and $P_D(s, \zeta)$, are given below.

$$
\boxed{P_A(s,b)} \qquad \max_{\substack{[A_{s\vartheta}(a)]\geq 0, [V_{ds'}],\\ [b_{ds'}(\vartheta)]\geq 0, V}} \; V
$$

$$
\text{s.t. } V \leq \sum_{a,\vartheta} A_{s\vartheta}(a) P^{\vartheta}_{sad} + \gamma \sum_{s'} V_{ds'} \quad \forall\, d
$$

$$
b_{ds'}(\vartheta) = \sum_{a} A_{s\vartheta}(a) T_{sad}(s') \quad \forall\, d, s', \vartheta
$$

$$
V_{ds'} \leq \Upsilon_v\big(\mathcal{Y}^A_s, \mathcal{W}^A_s, b_{ds'}\big) \quad \forall\, d, s'
$$

$$
\sum_{a} A_{s\vartheta}(a) = b(\vartheta) \quad \forall\, \vartheta
$$

$$
\boxed{P_D(s,\zeta)} \qquad \min_{\substack{[D_s(d)]\geq 0, [W_{as'}],\\ [\lambda_{as'}(\vartheta)]\geq 0, W}} \; W
$$

$$
\text{s.t. } W \geq \zeta(\vartheta) + \sum_{d} D_s(d) P^{\vartheta}_{sad}
$$

$$
+ \gamma \sum_{s'} \big(W_{as'} - \lambda_{as'}(\vartheta)\big) \quad \forall\, a, \vartheta
$$

$$
W_{as'} \geq \Upsilon_w\big(\mathcal{Y}^D_s, \mathcal{W}^D_s, \lambda_{as'}\big) \quad \forall\, a, s'
$$

$$
\sum_{d} D_s(d) = 1
$$

The objective functions of the two linear programs estimate the values of the primal and dual games. The decision variables $[A_{s\vartheta}], [V_{ds'}], [b_{ds'}(\vartheta)]$ in $P_A(s, b)$ are used to find the attacker's strategy, the continuation game value, and the updated belief, respectively. Similarly, $[D_s(d)], [W_{as'}], [\lambda_{as'}(\vartheta)]$

in $P_D(s, \zeta)$ are used to find the defender's strategy, the continuation game value, and the updated parameter ζ for the dual game, respectively. The first constraint in $P_A(s, b)$ encodes the defender's best response, replacing the minimization over $\bar{\pi}^D$ in Eq. (2). Similarly, the first constraint in $P_D(s, \zeta)$ replaces the maximization over μ in Eq. (5). The second and last constraints in $P_A(s, b)$ and the last constraint in $P_D(s, \zeta)$ enforce basic rules of probability. The third constraint in $P_A(s, b)$ and the second constraint in $P_D(s, \zeta)$ provide information state-value approximations of the continuation value estimates $V_{d,s'}$ and $W_{a,s'}$, respectively.

Due to convexity (concavity) of the value function v (w), we use the *sawtooth* function [11] as an example of this information state-value approximation. In particular, a lower bound on the primal game's value function $v(s, b)$ is given by $\Upsilon_v(\mathcal{Y}_s^A, \mathcal{W}_s^A, b_{ds'})$ whereas an upper bound of the dual game's value function $w(s, \zeta)$ is given by $\Upsilon_w(\mathcal{Y}_s^D, \mathcal{W}_s^D, \lambda_{as'})$.[4] The set $\mathcal{Y}_s^A$ contains the belief-value pairs associated with the beliefs that are non-corner points of the simplex over Θ for given s, whereas the set $\mathcal{W}_s^A$ contains the belief-value pairs associated with the corner points of the simplex. Analogously, $\mathcal{Y}_s^D, \mathcal{W}_s^D$ represent subsets of $\mathbb{R}^{|\Theta|}$ that contain the vectors with only one, and more than one, non-zero element, respectively. Details of both Υ_v and Υ_w using sawtooth functions can be found in the pseudocode in Sec. C in the Appendix.

Lemma 4 ensures that the sawtooth constraints are linear in the decision variables of the respective problem, which verifies the computational tractability of $P_A(s, b)$ and $P_D(s, \zeta)$. The proof of Lemma 4 can be found in Sec. B in the appendix.

Lemma 4. *By definitions of* SAWTOOTH-A *and* SAWTOOTH-D *in Algorithm 1, given $\mathcal{Y}_s, \mathcal{W}_s$, constraints $V_{ds'} \leq \Upsilon_v(\mathcal{Y}_s^A, \mathcal{W}_s^A, b_{ds'})$ and $W_{as'} \geq \Upsilon_w(\mathcal{Y}_s^D, \mathcal{W}_s^D, \lambda_{as'})$ are both* linear *in the decision variables $V_{ds'}, b_{ds'}$ and $W_{as'}, \lambda_{as'}$, respectively.*

Next, we numerically analyze the proposed algorithm in a cyber security environment.

5 Experiment: Intrusion Response

A recent trend in the security literature concerns the development of automated defense systems, termed state-based intrusion response systems, that automatically prescribe defense actions in response to intrusion alerts [26, 15, 27]. Core to these systems is the construction of a model that describes the possible ways an attacker can infiltrate the system, termed a *threat model*. Deriving a correct threat model is a challenging task and has a significant impact on the effectiveness of the intrusion response system. N-CIRL addresses one of the main challenges in this domain: the defender's uncertainty of the attacker's true intent.

The threat model in our experiment is based on an *attack graph*, a common graphical model in the security literature [2]. An attack graph is represented by a directed acyclic graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where each node $n \in \mathcal{N}$ represents a system condition and each edge $e_{ij} \in \mathcal{N} \times \mathcal{N}$ represents an exploit. Each exploit e_{ij} relates a *precondition* i, the condition needed for the attack to be attempted, to a *postcondition* j, the condition satisfied if the attack succeeds. Each exploit e_{ij} is associated with a probability of success, β_{ij}, describing the likelihood of the exploit succeeding (if attempted). The state space $\mathcal{S}$ is the set of currently satisfied conditions (enabled nodes). For a given state $s \in \mathcal{S}$, the attacker chooses among exploits that have enabled preconditions and at least one not yet enabled postcondition. The defender simultaneously chooses which exploits to block for the current stage; blocked edges have a probability of success of zero for the stage in which they are blocked. The attacker's reward is $R(s, a, d, s'; \theta) = r_e(s, s'; \theta) - c_A(a) + c_D(d)$, where s' is the updated state, $r_E(s, s'; \theta)$ is the attacker's reward for any newly enabled conditions, and $c_A(a)$ and $c_D(d)$ are costs for attack and defense actions, respectively. The experiments are run on many random instances of attack graphs; see some instances in Figure 1. See Sec. C for more details of the experimental setup.

As seen in Figure 2, the strategies obtained from N-CIRL yield a lower attacker reward than the strategies obtained from MA-IRL. Empirically, this implies that the defender benefits more from interleaving learning and execution than the attacker. Even though the interleaved setting may provide more ground for the attacker to exercise deceptive tactics, [36] states that in games of incomplete information, the more informed player "cannot exploit its private information without revealing it, at least to some extent." In the context of our example, we believe that the performance gain of

[4] Note that $b_{ds'}$ is the vector consisting of $b_{ds'}(\vartheta)$ over all $\vartheta \in \Theta$ (similarly for $\lambda_{as'}$).

N-CIRL arises from the fact that the attacker can only deceive for so long; eventually it must fulfill its true objective and, in turn, reveal its intent.

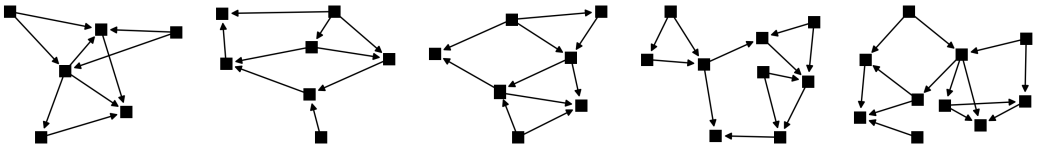

Figure 1: Instances of randomly generated graphs of sizes $n = 6$ to $n = 10$, with state cardinalities $|\mathcal{S}| = \{8, 24, 32, 48, 64\}$ and action cardinalities $|\mathcal{A}| = |\mathcal{D}| = \{54, 88, 256, 368, 676\}$.

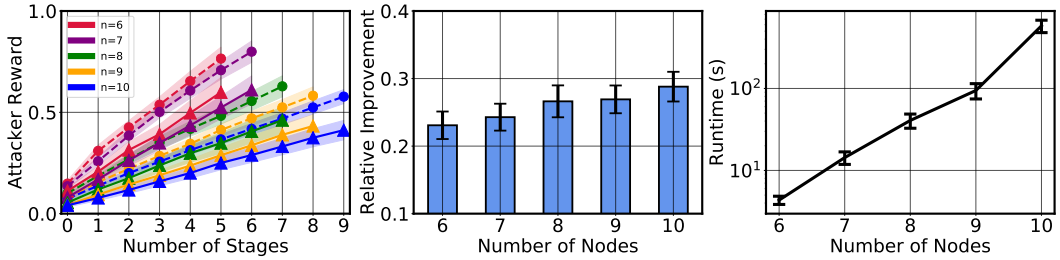

Figure 2: *Left*: Attacker's average reward for each graph size n. Dotted and solid lines represent attacker's value against defense strategies computed under MA-IRL and N-CIRL, respectively. *Middle*: The average relative reduction of attacker reward in N-CIRL compared to MA-IRL as a function of n. *Right*: Average runtime (in seconds) as a function of n.

6 Concluding Remarks

The goal of our paper was to introduce the N-CIRL formalism and provide some theoretical results for the design of learning algorithms in the presence of strategic opponents. The primary motivation for this work was cyber security, specifically, problems where the defender is actively trying to defend a network when it is uncertain of the attacker's true intent. Learning from past attack traces (*i.e.*, equilibrium behavior/demonstrations) can lead to poor defense strategies, demonstrating that such approaches (*e.g.*, MA-IRL) are not directly applicable for settings where rewards may change between demonstrations. Empirical studies illustrate that the defender can benefit by interleaving the learning and execution phases compared to just learning from equilibrium behavior (this is an analogous conclusion to the one found in CIRL that an interactive scenario can lead to better performance). The reason for this is that defense strategies computed using N-CIRL learn the intent adaptively through interaction with the attacker.

As shown in [9], the value alignment problem is more appropriately addressed in a dynamic and cooperative setting. The cooperative reformulation converts the IRL problem into a decentralized stochastic control problem. In our paper, we have shown that the non-cooperative analog of CIRL, *i.e.*, when agents possess goals that are misaligned, becomes a zero-sum Markov game with one-sided incomplete information. Such games are conceptually challenging due to the ability of agents to influence others' beliefs of their private information through their actions, termed *signaling* in game theoretic parlance. We hope that the N-CIRL setting can provide a foundation for an algorithmic perspective of these games and deeper investigation into signaling effects in general stochastic games of asymmetric information.

Acknowledgements

This work was supported in part by the US Army Research Laboratory (ARL) Cooperative Agreement W911NF-17-2-0196, and in part by the Office of Naval Research (ONR) MURI Grant N00014-16-1-2710.

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, 2004.

[2] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *ACM Conference on Computer and Communications Security*, 2002.

[3] T. Arnold, D. Kasenberg, and M. Scheutz. Value alignment or misalignment – what will keep systems accountable? In *AAAI Conference on Artificial Intelligence*, 2017.

[4] T. Başar. Stochastic differential games and intricacy of information structures. In *Dynamic Games in Economics*, pages 23–49. Springer, 2014.

[5] D. Blackwell. Discounted dynamic programming. *The Annals of Mathematical Statistics*, 36(1):226–235, 1965.

[6] P. Cardaliaguet and C. Rainer. Stochastic differential games with asymmetric information. *Applied Mathematics and Optimization*, 59(1):1–36, 2009.

[7] B. De Meyer. Repeated games, duality and the central limit theorem. *Mathematics of Operations Research*, 21(1):237–251, 1996.

[8] C. Finn, S. Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, 2016.

[9] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.

[10] E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI Conference on Artificial Intelligence*, 2004.

[11] M. Hauskrecht. Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 13:33–94, 2000.

[12] K. Hendricks and R. P. McAfee. Feints. *Journal of Economics & Management Strategy*, 15(2):431–456, 2006.

[13] K. Horák, B. Bosanskỳ, and M. Pechoucek. Heuristic search value iteration for one-sided partially observable stochastic games. In *AAAI Conference on Artificial Intelligence*, 2017.

[14] J. Hörner, D. Rosenberg, E. Solan, and N. Vieille. On a Markov game with one-sided information. *Operations Research*, 58(4-part-2):1107–1115, 2010.

[15] S. Iannucci and S. Abdelwahed. A probabilistic approach to autonomic security management. In *IEEE International Conference on Autonomic Computing*, 2016.

[16] Y. Ji, S. Lee, E. Downing, W. Wang, M. Fazzini, T. Kim, A. Orso, and W. Lee. RAIN: Refinable attack investigation with on-demand inter-process information flow tracking. In *ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[17] R. E. Kalman. When is a linear control system optimal? *Journal of Basic Engineering*, 86(1):51–60, 1964.

[18] H. W. Kuhn. *Extensive Games and the Problem of Information*, volume 2. Princeton University Press, 1953.

[19] M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, and T. Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

[20] X. Lin, S. C. Adams, and P. A. Beling. Multi-agent inverse reinforcement learning for general-sum stochastic games. *arXiv preprint arXiv:1806.09795*, 2018.

[21] X. Lin, P. A. Beling, and R. Cogill. Multiagent inverse reinforcement learning for two-person zero-sum games. *IEEE Transactions on Games*, 10(1):56–68, 2018.

[22] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings*, pages 157–163. Elsevier, 1994.

[23] D. Malik, M. Palaniappan, J. Fisac, D. Hadfield-Mennell, S. Russell, and A. Dragan. An efficient, generalized Bellman update for cooperative inverse reinforcement learning. In *International Conference on Machine Learning*, 2018.

[24] J. Marschak and R. Radner. *Economic Theory of Teams.* Yale University Press, 1972.

[25] J.-F. Mertens and S. Zamir. Formulation of bayesian analysis for games with incomplete information. *International Journal of Game Theory*, 14(1):1–29, 1985.

[26] E. Miehling, M. Rasouli, and D. Teneketzis. Optimal defense policies for partially observable spreading processes on Bayesian attack graphs. In *Second ACM Workshop on Moving Target Defense*, 2015.

[27] E. Miehling, M. Rasouli, and D. Teneketzis. A POMDP approach to the dynamic defense of large-scale cyber networks. *IEEE Transactions on Information Forensics and Security*, 13(10):2490–2505, 2018.

[28] A. Nayyar and T. Başar. Dynamic stochastic games with asymmetric information. In *IEEE Conference on Decision and Control*, 2012.

[29] A. Nayyar, A. Gupta, C. Langbort, and T. Başar. Common information based Markov perfect equilibria for stochastic games with asymmetric information: Finite games. *IEEE Transactions on Automatic Control*, 59(3):555–570, 2014.

[30] A. Nayyar, A. Mahajan, and D. Teneketzis. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013.

[31] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, 2000.

[32] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conferences on Artificial Intelligence*, 2003.

[33] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. Maximum margin planning. In *International Conference on Machine Learning*, 2006.

[34] J. Renault. The value of Markov chain games with lack of information on one side. *Mathematics of Operations Research*, 31(3):490–512, 2006.

[35] D. Rosenberg. Duality and Markovian strategies. *International Journal of Game Theory*, 27(4):577–597, 1998.

[36] D. Rosenberg and N. Vieille. The maxmin of recursive games with incomplete information on one side. *Mathematics of Operations Research*, 25(1):23–35, 2000.

[37] S. Russell, D. Dewey, and M. Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI Magazine*, 36(4):105–114, 2015.

[38] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Malaysia; Pearson Education Limited, 2016.

[39] S. Sorin. Stochastic games with incomplete information. In *Stochastic Games and Applications*, pages 375–395. Springer, 2003.

[40] S. Srinivasan, M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, and M. Bowling. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems*, 2018.

[41] X. Wang and D. Klabjan. Competitive multi-agent inverse reinforcement learning with sub-optimal demonstrations. In *International Conference on Machine Learning*, 2018.

[42] N. Wiener. Some moral and technical consequences of automation. *Science*, 131(3410):1355–1358, 1960.

[43] K. Zhang, Z. Yang, and T. Başar. Networked multi-agent reinforcement learning in continuous spaces. In *IEEE Conference on Decision and Control*, 2018.

[44] K. Zhang, Z. Yang, and T. Başar. Policy optimization provably converges to Nash equilibria in zero-sum linear quadratic games. *arXiv preprint arXiv:1906.00729*, 2019.

[45] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, 2018.