PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Adversary decision-making using Markov models

Elizabeth Andreas, Jessica Dorismond, Marco Gamarra

Elizabeth Andreas, Jessica Dorismond, Marco Gamarra, "Adversary decision-making using Markov models," Proc. SPIE 12538, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications V, 125380I (12 June 2023); doi: 10.1117/12.2655223

SPIE.

Event: SPIE Defense + Commercial Sensing, 2023, Orlando, Florida, United States

Adversary Decision-making using Markov Models

Elizabeth Andreas[a], Jessica Dorismond[b], and Marco Gamarra[c]

[a]Department of Mathematics, Montana State University-Bozeman, Bozeman, MT,USA
[b,c]Air Force Research Lab, Rome, NY, USA

ABSTRACT

This study conducts three experiments on adversary decision-making modeled as a graph. Each experiment has the overall goal to understand how to exploit an adversary's decision-making in order to obtain desired outcomes, as well as specific goals unique to each experiment. The first experiment models adversary decision-making using an Absorbing Markov chain (AMC). A sensitivity analysis of states (nodes in the graph) and actions (edges in the graph) is conducted which informs how downstream adversary decisions could be manipulated. The next experiment uses a Markov decision process (MDP). Assuming the adversary is initially blind to the rewards they will receive when they take an action, a $Q-$learning algorithm is used to determine the sequence of actions that maximizes the adversary rewards (called an optimum policy). This experiment gives insight in the possible decision-making of an adversary. Lastly, in the third experiment a two-player Markov game is developed, played by an agent (friend) and the adversary (foe). The agents goal is to decrease the overall rewards the adversary receives when it follows optimum policy. All experiments are demonstrated using specific examples.

Introduction

To have a strategic decision advantage over our adversaries is vital to have the ability to convince or deter adversaries from taking actions that threaten the interests of the United States and its military. This paper will explore if a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event can be a potential tool to shape an adversary decision calculus for a specific operational context. To demonstrate the usefulness of this approach, we applied this technique to an operational vignette based on the Air Force Concept of Agile Combat Employment (ACE). The ACE vignette refer to operating in an adaptive manner from a fluid set of basing locations to render the adversary targeting of US forward forces more complex. For this vignette, the target of the attack is the adversary's decision calculus and the adversary's assessment of whether an offensive first strike is likely to deny Blue's freedom of movement.

This study conducts three different experiments using stochastic models that could be used to advise the United States Air Force (USAF) on how to influence adversary decision-making. Each mathematical model depicts adversary decision-making as a graph. The nodes of the graph are the different adversary **states** and the edges are the feasible **actions** the adversary can take. The goal of each experiment is to understand how an agent (the USAF or a friend) could exploit adversary's decision-making to obtain desired outcomes for the agent. The two major assumptions for this research are that the adversary has at least one end goal and that the mathematical models have the Markov property.

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

The first experiment aims to determine which states and actions are critical to the adversary by modeling the adversary's decision calculus using an Absorbing Markov chain (AMC). This experiment was conducted by performing a sensitivity analysis of the AMC. The second experiment models the adversary decision calculus as a Markov decision process (MDP). The purpose of the second experiment is to understand what sequence of states and actions the adversary is likely to take when each action has a perceived reward. For this model, we will assume the adversary is initially blind to the rewards they will receive when they take an action, and a $Q-$learning algorithm is used to determine the sequence of actions that maximizes the adversary rewards (called an optimum policy). Finally, the last experiment aims to extend the results of the MDP by determining if an agent can deter the adversary from its original optimum policy in such a way that the adversary obtains fewer rewards. This is done using a two-player Markov game played by the agent and the adversary.

Experiments

We begin our analysis by laying out the adversary decision calculus as it applies to the ACE scenario. The adversary decision calculus represents the base state of adversarial decision-making and Blue operations and the attempt to shape the outcome of the adversary decision flow to achieve a desired set of operational effects[3]. Three main experiments exploring adversary decision-making were conducted. For each, the adversary's decision-making process was modeled as a graph $G = \{S, A\}$, where S is the set of states and A is the set of actions. Each $s_i \in S$ is a node in G which represents an event and each $(s_i, s_j) \in A$ is an edge in G which represents the action the adversary takes, transitioning them from state s_i to state s_j. A state that is impossible to leave is called an **absorbing state** and a **transient state** is any state that is not absorbing [2]. The graph shown in Figure 1 was used for all experiments and will be referenced often, thus we will denote it as G_{adv}. Note that while not shown in Figure 1, all states have a self-loop which is the *do nothing* action.

Experiment 1: Absorbing Markov Chain

In this experiment, the adversary decision-making is analyzed assuming the following:

1. every state the adversary can be in is known,
2. every probability of an adversary taking an action is known and
3. past events have no affect on future events (known as a **Markov property**).

The goals are to find what states the adversary is likely to be in, what absorbing state the adversary is likely to end in when starting at state s_1 (state 1 in G_{adv}), and how the decision-making process can be manipulated to give more desirable results for the agent. These goals can be reached using a Markov chain model, which is now described.

Specifically, the Markov property says that the probability of transitioning from state $s_i \in S$ to state $s_j \in S$ only depends on s_i. A **Markov chain** is a model describing an order of possible events where each step has the Markov property [7]. An **absorbing Markov chain** is a Markov

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

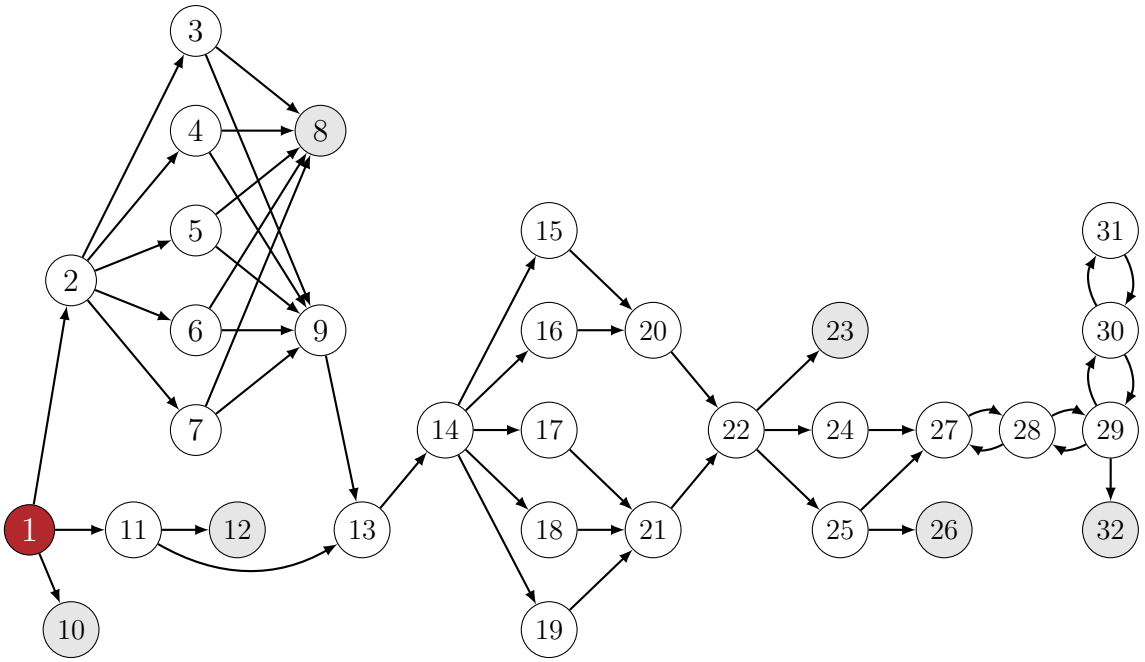

Figure 1: Adversary decision-making graph model for experiments. Each state has a *do nothing* action (self-loop) which is not shown for simplicity. Absorbing states are shown in gray, and the starting node is shown in red.

chain where each state can reach an absorbing state in a finite number of steps [2][7]. An absorbing Markov chain can be represented as a **transition matrix** [7]

$$P = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{n,1} & \cdots & p_{n,n} \end{bmatrix}$$

where $n = \text{card}(S)$, the $p_{i,j}$ entry is the probability of transitioning from state s_i to state s_j and

$$\sum_{j=1}^{n} p_{i,j} = 1 \tag{1}$$

for each $i = 1, ..., n$ [7]. For example, consider the graph G_{adv} in Figure 1 and let s_i denote state i in G_{adv}. Given the following probabilities $p_{1,1} = 0.33$, $p_{1,2} = 0.52$, $p_{1,10} = 0.13$, $p_{1,11} = 0.02$ and $p_{1,j} = 0$ for all $j \in \{3, ..., 9, 12, ..., 32\}$ of transitioning from state s_1, we have

$$\sum_{j=1}^{n} p_{1,j} = 0.33 + 0.52 + 0.13 + 0.02 = 1.$$

Further, we can transform the transition matrix P in canonical form where

$$P = \begin{bmatrix} Q_{t\times t} & R_{t\times r} \\ \mathbf{0}_{r\times t} & I_{r\times r} \end{bmatrix}$$

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

where t is the number of transient states, r is the number of absorbing states, $\mathbf{0}_{r\times t}$ is the zero matrix and $I_{r\times r}$ is the identity matrix [2][7]. $Q_{t\times t}$ then describes the probability of transitioning from some transient state to another and $R_{r\times r}$ describes the probability of transitioning from some transient state to an absorbing state. This transformation is convenient as the probability of transitioning from transient state s_i to another transient state s_j in k steps is the $(i,j)-$th entry of Q^k [7]. Then the expected number of times a transient state s_j is visited from s_i is the $(i,j)-$th entry of

$$N = \sum_{k=1}^{\infty} Q^k = (1-Q)^{-1}, \tag{2}$$

where N is called the **fundamental matrix** of the absorbing Markov chain [7]. Lastly, the (i,j)-th entry of the matrix

$$B = NR \tag{3}$$

gives the probability of transitioning from transition state s_i to absorbing state s_j [7].

Additionally, let a **critical increasing (decreasing) state** of state s_i be some state $s_j \neq s_i$ such that removing s_j from G results in the highest increase (decrease) in the probability of ending on state s_i and s_j is not the starting state. Let a **critical increasing (decreasing) action** of state s_i be some action $(s_j, s_k) \in A$ such that adjusting the probability of taking action (s_j, s_k) results in the highest increase (decrease) in the probability of ending on state s_i.

Experiment 2: Markov Decision Process

In this experiment, instead of having a probability of an adversary transitioning from one state to another, each action $(s_i, s_j) \in A$ is assigned a **reward** $r_{i,j}$ equivalent to the adversary perception of gains when taking (s_i, s_j). Let R be the **reward matrix** with entries $r_{i,j}$. The goal of this experiment is to predict the optimum order of events that maximizes the overall reward the adversary can receive, called an **optimum policy** [6]. A **policy** $\pi(s_j|s_i)$ (at state s_i) is some rule for deciding what action to take given the current state s_i [6]. There are two different types of polices, stationary and stochastic. A stationary policy specifies that the same action be taken at every state while a stochastic policy specifies that each action is chosen independently according to the current state [8]. From now on, a policy π will be referring to a stochastic policy with the additional conditions that π is finite, begins at s_1 and ends in an absorbing state. Specifically, π is a path in G such that

$$\pi = s_{i_1} \rightarrow s_{i_2} \rightarrow ... \rightarrow s_{i_k}$$

where $k \in \mathbb{Z}$ is finite, s_{i_1} is the starting state and s_{i_k} is an absorbing state. Further, the optimum policy π^* is some π that with the maximum expected reward [8].

If the adversary knows the reward of each action, then π^* is simply the policy π where each step is maximizing the reward. However, in this experiment requires the additional assumption that the adversary is blind to the rewards and therefore must learn the value of each state by exploring the graph. Then with the following assumptions,

1. every state the adversary can be in is known,
2. every action is known,

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

3. every reward the adversary perceives they will obtain when taking an action is known by the model but not the adversary and

4. past events have no affect on future events,

a **Markov decision process** (MDP) can be used to find π^* for the adversary. A MDP is a mathematical model, defined by the tuple $\{S, A, R\}$ [1][5], and is described below.

Let $Q(s_i, s_j)$ be the **state-action value**, which is the expected reward obtained by the adversary when they start at state s_i, go to state s_j and follows π^* afterwards. Initially, $Q(s_i, s_j) = 1$ for all $i = 1, ..., \text{card}(S)$ and $j = 1, ..., \text{card}(S)$, then is updated during an exploration stage using the Q-learning algorithm

$$Q(s_i, s_j) \leftarrow Q(s_i, s_j)(1 - \ell) + \ell(r_{i,j} + \gamma \max_{s'_k \in A} Q(s_j, s'_k)) \tag{4}$$

where A' is the set of states s'_k such that $(s_j, s'_k) \in A$, $0 \leqslant \ell \leqslant 1$ is called the **learning rate** (decayed after each action by some $0 \leqslant \text{DECAY} \leqslant 1$) and $0 \leqslant \gamma \leqslant 1$ is called the **decay rate** [1][5][8]. If $\ell = 0$ then the adversary only remembers the current $Q(s_i, s_j)$, if $\ell = 1$ then the adversary only learns the updated $Q(s_i, s_j)$. As $\gamma \to 0$, the more immediate reward driven the adversary is [1]. Let Q be the matrix where the $(i, j)-$th entry is $Q(s_i, s_j)$ and let ε-policy be some π. The procedure for assigning each step in the ε-policy is done by randomly choosing an action at each state according to the probabilities

$$p(s_j|s_i) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{\text{card}(A')} & \text{if } s_j = \text{argmax}_{s'_j \in A'} Q(s_i, s'_j) \\ \frac{\varepsilon}{\text{card}(A')} & \text{otherwise} \end{cases} \tag{5}$$

where A' is the set of states s'_j such that $(s_i, s'_j) \in A$ until an absorbing state is reached [6]. Notice that if $\varepsilon = 1$, then the probably of any action in A' is $\frac{\varepsilon}{\text{card}(A')}$. As $\varepsilon \to 0$, the more greedy the adversary is exploring, i.e., the adversary is more likely to take an action if they believe they are going to get higher rewards.

During the exploration stage, the adversary is following some ε-policy until entries of Q for each step in the ε-policy remains constant, then the adversary follows a new ε-policy until entries of Q for each step in the ε-policy remains constant. This repeats until Q remains constant for any ε-policy. After the exploration stage, the optimum policy if found, from the starting state, by choosing the action $(s_i, s_j) \in A$ at each step where $(s_i, s_j) = \text{argmax}_{s'_j \in A'} Q(s_i, s'_j)$, with

$$Q(s_i, s_j) \leftarrow \gamma Q(s_i, s_j)$$

every time $s_i = s_j$. This discount by γ is applied because it is assumed that the longer an adversary is in a state, the less rewarding it becomes, this assumption also helps prevent infinite policies. Formally, Algorithm 2 in Appendix describes this entire process.

Experiment 3: Markov Game

The last experiment is a **Markov game** (MG) and extends the MDP experiment from a one-player model to a two-player game. Suppose we have two players, an agent who is a friend and an adversary who is a foe. Furthermore, suppose that the adversary decision-making process is still modeled by the graph G. The goal of the agent is to influence the decision-making process of the adversary.

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

For this experiment, the agent works towards their goal by attempting to minimize the maximum reward the adversary can obtain. Thus, for each state $s_i \in S$ the agent will have a set of actions $O_i = \{o_{i,j_1}, ..., o_{i,j_q}\}$ for finite $q \in \mathbb{Z}$ that impacts the reward the adversary can receive. Then the entire action set for the agent is $O := \bigcup_i O_i$ for $i \in \{1, ..., \text{card}(S)\}$. Specifically, the agent will have an action $o_{i,j}$ for each $(s_i, s_j) \in A$. For example, suppose that the adversary is in the starting state $s_1 \in G_{adv}$, then the adversary can take one of the four actions $(s_1, s_1), (s_1, s_2), (s_1, s_{10})$ and (s_1, s_{11}). Corresponding to each adversary action the agent has actions $o_{1,1}$, $o_{1,2}$, $o_{1,10}$, $o_{1,11}$ respectively that only affect the rewards received by the adversary. See Figure 2 for a depiction of this example. Further, this game will be turned based where the agent chooses an action before the adversary does.

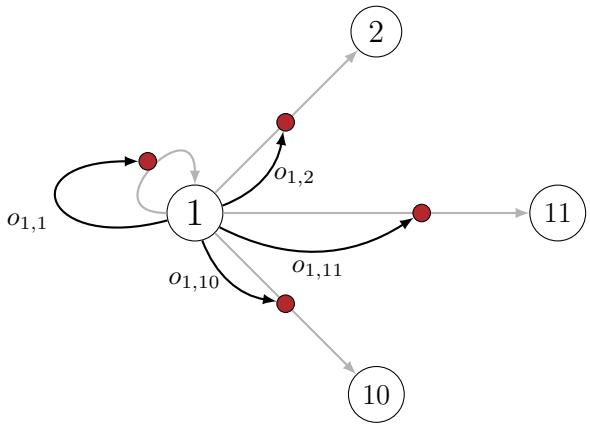

Figure 2: An example of actions taken by an agent when the adversary is at state $s_1 = 1 \in G_{adv}$. The states (white filled nodes) represent the corresponding states in G_{adv} and the gray edges are the possible actions that the adversary can take from state s_1. The black edges and red filled nodes are the possible actions the agent can take, given the adversary is state s_1.

Let $R_{adv}(s_i, s_j, o)$ be the reward the adversary receives when they start at state s_i and go to state s_j when the agent took action $o \in O_i$. Let $Q_{adv}(s_i, s_j, o)$ be the state-action value for the adversary when they start at state s_i and go to state s_j when the agent takes action $o \in O_i$. The adversary's goals are the same as in the MDP experiment, thus we will use the two-player Q-learning algorithm [5][9]

$$Q_{adv}(s_i, s_j, o) \leftarrow Q_{adv}(s_i, s_j, o)(1 - \ell) + \ell(R_{adv}(s_i, s_j, o) + \gamma \max_{s'_k \in A'} Q_{adv}(s_j, s_k, o')) \tag{6}$$

where $o' \in O_j$. The optimum policy π^*_{adv} for the adversary is determined the same as in the MDP experiment. If all of the agents actions have no affect on the rewards the adversary receives, then

$$Q_{adv}(s_i, s_j, o) = Q(s_i, s_j), \text{ and } \pi^*_{adv} = \pi^*$$

where Q and $\pi*$ are the state action value matrix and optimum policy from the MDP experiment. For this game, this will be called the **base case** and is given a score q_{MDP}. This score is calculated while computing the optimum policy, where at step t of π^*,

$$q_{\text{MDP}} \leftarrow \begin{cases} q_{\text{MDP}} + Q(s_t, s_{t+1}) & \text{if } s_t \neq s_{t+1} \\ q_{\text{MDP}} + \gamma^{n_t} Q(s_t, s_{t+1}) & \text{if } s_t = s_{t+1} \end{cases}$$

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

where n_t is the number of times state s_t has shown up in π^* so far. Repeated states are discounted under an assumption that it becomes more costly for the adversary to stay in one state for an extended amount of time. This also helps prevent infinite optimum policies.

Now the **operation case** begins, let $Q_{agt}(s_i, s_j, o)$ be the state-action value for the agent when they choose action $o \in O_i$, and the adversary goes from state s_i to state s_j, we can compute $Q_{agt}(s_i, s_j, o)$ by

$$Q_{agt}(s_i, s_j, o) \leftarrow Q_{agt}(s_i, s_j, o)(1 - \ell) + \ell(-R_{adv}(s_i, s_j, o) + \gamma \min_{o' \in O_j} \max_{s'_k \in A'} Q_{adv}(s_j, s'_k, o')) \tag{7}$$

where A' is the set of states s'_k such that $(s_j, s'_k) \in A$. Notice that instead of defining a different reward function for the agent, that Equation7 uses $-R_{adv}(s_i, s_j, o)$. This is because the agent's goal is to minimize the maximum reward the adversary can receive, so whenever the adversary obtains a high reward, the agent didn't reach its goals and is negativity impacted. Equation 7 is an adaption of the minimax-Q equation in [4] and [5]. Note that the agent is not attempting to traverse the graph like the adversary is. Thus, the optimum policy π^*_{agt} is a sequence of actions corresponding to π^*_{adv} and not a path in G_{adv} like π^*_{adv} is.

Let Q_{agt} and Q_{adv} be the state action value matrices for the agent and adversary respectively. As in the MDP experiment, there is an exploration stage where both are following the $\varepsilon-$policy, but where the $\varepsilon-$policy is assigned in turns, i.e., from state s_i the agent is assigned random action $o' \in O_i$ according to the probabilities

$$p(o|s_i) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{\text{card}(O_i)} & \text{if } o = \text{argmin}_{o \in O_i} \max_{s'_j \in A'} Q(s_i, s'_j, o) \\ \frac{\varepsilon}{\text{card}(O_i)} & \text{otherwise} \end{cases} \tag{8}$$

where A' is the set of states s'_j such that $(s_i, s'_j) \in A$ [6]. Then the adversary takes a random step according to probabilities [6]

$$p(s_j|s_i) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{\text{card}(A')} & \text{if } s_j = \text{argmax}_{s'_j \in A'} Q(s_i, s'_j, o') \\ \frac{\varepsilon}{\text{card}(A')} & \text{otherwise.} \end{cases} \tag{9}$$

There is still an assumption that any $\varepsilon-$policy must begin at the starting state and end in an absorbing state. During the exploration stage, the agent and adversary follow some ε-policy until entries of Q_{agt} and Q_{adv} for each step in the ε-policy remains constant, then the adversary follows a new ε-policy until entries of Q_{agt} and Q_{adv} for each step in the ε-policy remains constant. This repeats until Q_{agt} and Q_{adv} remains constant for any ε-policy. After the exploration stage, π^*_{agt} and π^*_{adv} are computed in turns. Informally the turns go as follows, from state s_i and with score q_{MG},

1. the agent takes the action $o \in O_j$ such that $o = \text{argmin}_{o' \in O_i} \max_{s'_j \in A'} Q_{adv}(s_i, s'_j, o')$, then
2. the adversary takes action (s_i, s_j) such that $s_j = \text{argmax}_{s'_j \in A'} Q_{adv}(s_i, s'_j, o)$,
3. if $s_i = s_j$ then $Q_{adv}(s_i, s'_j, o) \leftarrow \gamma Q_{adv}(s_i, s'_j, o)$ for all $o \in O_i$,
4. $q_{\text{MG}} \leftarrow q_{\text{MG}} + Q_{adv}(s_i, s'_j, o)$.

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

Now, the results of the game are

- agent wins if $q_{\text{MG}} < q_{\text{MDP}}$,
- agent loses if $q_{\text{MG}} > q_{\text{MDP}}$,
- tie if $q_{\text{MG}} = q_{\text{MDP}}$, and
- draw if the game has policy that doesn't reach an absorbing state before some amount of time.

Formally, Algorithm 4 in Appendix describes this entire process.

Results

Experiment 1: Absorbing Markov Chain

Until stated otherwise, the probabilities of the adversary transitioning from one state to another were fixed and can be found in Appendix 2 for results recreation. Recall that given the transition matrix of G_{adv}, it is possible to determine the probabilities of the adversary arriving at any state s_j from the starting state s_1 in k steps. The results for s_j being one of the transient states s_1, s_2, s_6 or s_{13} are show in Figure 3 (left) along with the results for s_j being one of the absorbing states (right). Using Equation 3 to compute B for G_{adv}, the probabilities of transitioning from the starting state

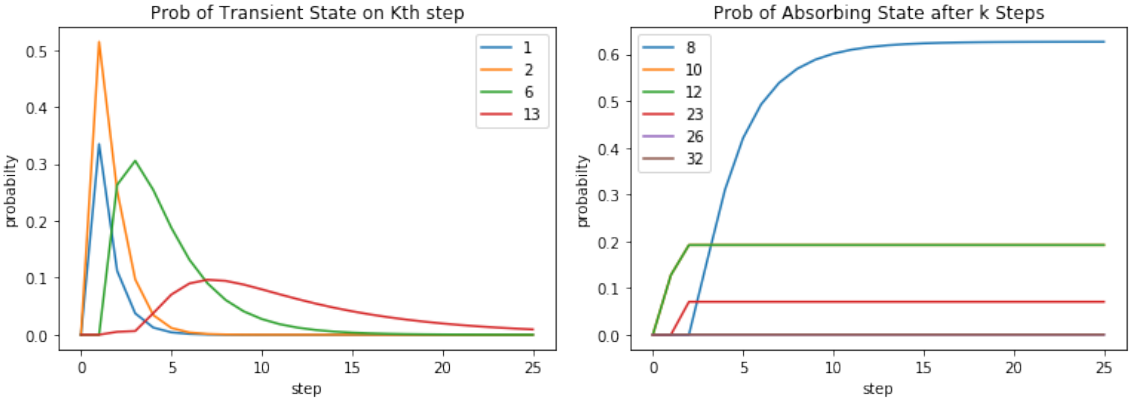

Figure 3: Left: probability of adversary being in state s_1, s_2, s_6 or s_{13} on the $k-$th step when starting from state s_1. Right: probability of adversary being in state an absorbing state on the $k-$th step when starting from state s_1.

s_1 to some absorbing state s_j for $j \in \{8, 10, 12, 23, 26, 32\}$ were $p(s_8|s_1) = 0.63$, $p(s_{10}|s_1) = 0.19$, $p(s_{12}|s_1) = 0.03$, $p(s_{23}|s_1) = 0.07$, $p(s_{26}|s_1) = 0$ and $p(s_{32}|s_1) = 0.08$. This can also be found in Table 1 for easy reference. It was found that the critical increasing states for each absorbing state $s_8, s_{10}, s_{12}, s_{23}, s_{26}$ and s_{32} were states $s_{10}, s_2, s_2, s_8, s_{27}$ and s_8 respectively. These can also be seen in Table 1, along with the new probabilities of ending in the absorbing state. The critical decreasing state was always one that disconnected the absorbing state from s_1, aside from absorbing state s_{10}. Absorbing state s_{10} didn't have a critical decrease state as only removing states s_1 and s_{10} decreased the chances of being in s_{10} but these cannot be critical decreasing states by definition.

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

It was found that the critical increasing actions for each absorbing state $s_8, s_{10}, s_{12}, s_{23}, s_{26}$ and s_{32} were actions $(s_1, s_2), (s_1, s_{10}), (s_1, s_{11}), (s_2, s_4), (s_{25}, s_{26})$, and (s_2, s_4) respectively and the critical decreasing actions were (s_1, s_2), (s_1, s_{11}), (s_{11}, s_{12}), (s_{22}, s_{23}), (s_{22}, s_{25}), and (s_1, s_{10}). All of these results can also be seen in Table 1.

Absorbing State:	s_8	s_{10}	s_{12}	s_{23}	s_{26}	s_{32}
Original Probability from s_1:	0.63	0.19	0.03	0.07	0.0	0.08
Critical State (inc):	s_{10}	s_2	s_2	s_8	s_{27}	s_8
New Probability from s_1:	0.78	0.85	0.12	0.36	0.05	0.42
Critical Action (inc):	(s_1, s_2)	(s_1, s_{10})	(s_1, s_{11})	(s_2, s_4)	(s_{25}, s_{26})	(s_2, s_4)
Critical Action (dec):	(s_1, s_2)	(s_1, s_{11})	(s_{11}, s_{12})	(s_{22}, s_{23})	(s_{22}, s_{25})	(s_1, s_{10})

Table 1: For each absorbing state in G_{adv}, the probability of the adversary ending is shown before and after removing the absorbing states critical increasing state. Further, the critical increasing and decreasing action is shown for each absorbing state.

Notice that absorbing state s_8 has the same increasing and decreasing critical action. A visual of how the probability of ending on s_8 is affected by either increasing or decreasing the probability of taking action (s_1, s_2) is shown in the left graph of Figure 4. The dotted line is the original probability of ending on state s_8 and the blue line is showing how that probability changes as the probability of taking action (s_1, s_2) is increased from 0 to 1. This graph shows that to increase the probability of ending on state s_8, the probability of taking action (s_1, s_2) need to be increased while decreasing the probability of taking action (s_1, s_2) decreases the chances of ending on state s_8. The middle and right graphs are showing the critical increasing and decreasing actions for absorbing state s_{32}.

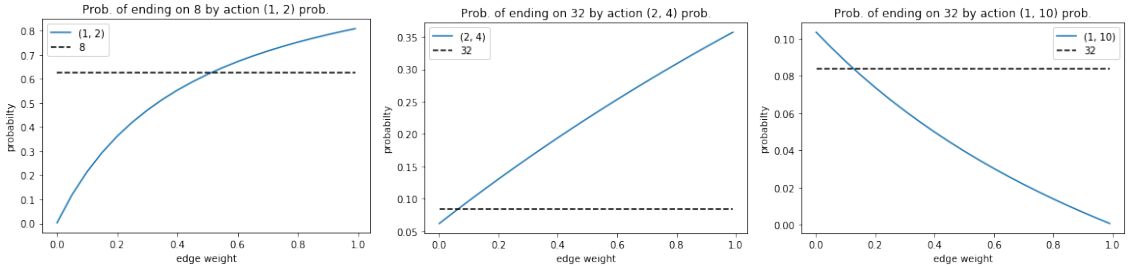

Figure 4: Left: critical increasing and decreasing action results for s_8 as the probability of taking action (s_1, s_2) is changed. Middle: critical increasing action results for s_{32} as the probability of taking action (s_2, s_4) is changed. Right: critical decreasing action results for s_{32} as the probability of taking action (s_1, s_{10}) is changed. Dotted lines are the original probabilities of the absorbing state and the blue lines are the new probabilities of ending on the absorbing state given the probability of the action.

Lastly, suppose that the probability of the adversary taking each action isn't know. Then a critical state and action analysis can be done using a simulation run multiple times that repeats the following steps

1. assign probabilities to each action randomly,

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

2. for each absorbing state, determine the critical states and actions.

Some results of this simulation can be seen in Figure 5 for absorbing state s_8.

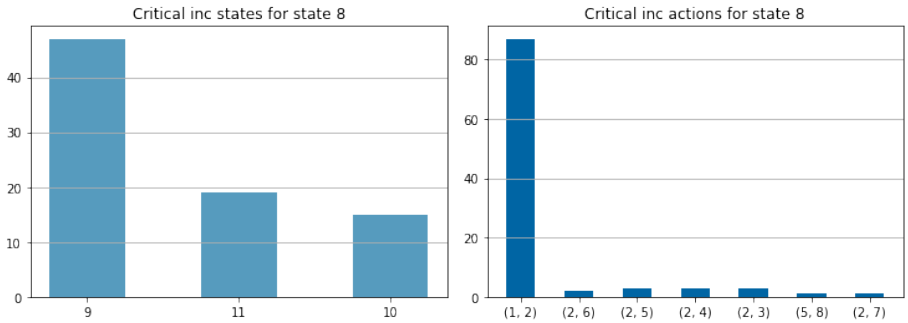

Figure 5: Left: critical increasing state results for absorbing state s_8. Each bar is the number of times that state was the critical increasing state out of 100 simulation runs. Right: critical increasing action results for absorbing state s_8. Each bar is the number of times that action was the critical increasing action out of 100 simulation runs.

Experiment 2: Markov Decision Process

This analysis was done for a fixed reward matrix R which can be found in Appendix 2 for recreating the results. This section shows results for what happens to the optimum policy π^* when one of the following are adjusted

- the decay rate γ,
- ε for exploration using $\varepsilon-$policy, and
- DECAY for learning rate ℓ.

First, let $\varepsilon = 1$ and DECAY $= 1$ while γ is set to values between 0 and 1. Figure 6 shows the optimum policies when $0 \leqslant \gamma < 0.34$ (yellow), $0.34 \leqslant \gamma < 0.55$ (red) and $0.55 \leqslant \gamma < 0.75$ (blue). For all $0.75 \leqslant \gamma \leqslant 0.99$, the optimum policy ends in absorbing state s_{32} but the policy lengths were between 59 and 316 steps long, jumping between states $s_{27}, s_{28}, s_{29}, s_{30}$, and s_{31} multiple times before ending on s_{32}. The case for $\gamma = 1$ was not computed as the state-value matrix Q didn't stop changing during the exploration stage.

Now, fix $\gamma = 0.5$ and DECAY $= 1$ while ε is set to values between 0 and 1. Recall that when $\varepsilon = 1$, that the probability of taking any action (s_i, s_j) from state s_i are the same. Then as $\varepsilon \to 0$, the probability of taking the action with the highest state-value becomes more likely. In this example, for each ε the optimum policy didn't change. This is good, as the exploration state persists until Q converges, there is no expectation that that the optimum policy would change as $\varepsilon \to 0$. However, the exploration stage takes longer for ε closer to 0. Further, the amount of times states were visited changes. For example, when the number of times the graph is explored during the exploration stage was fixed to 1000, states s_2, s_7, s_{15} and s_{31} were visited $390, 110, 147$ and 315 times respectively when $\varepsilon = 1$ and $2327, 2214, 38$ and 49 times respectively when $\varepsilon = 0.1$.

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

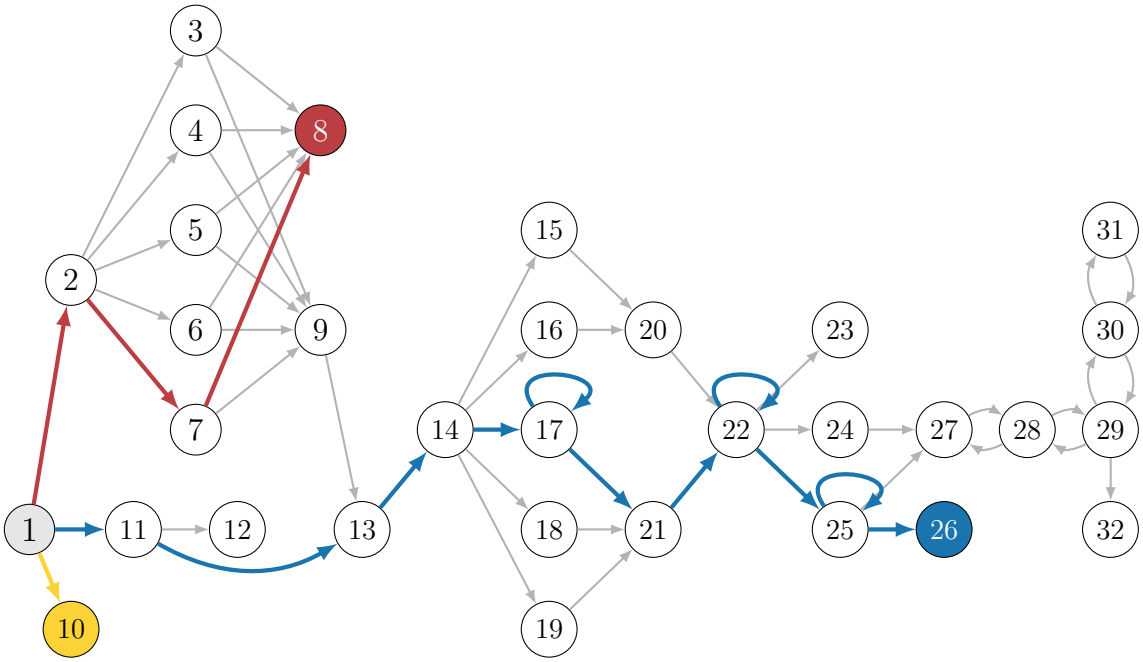

Figure 6: Adversary decision-making graph model with each colored path an optimum policy associated to some γ and all other variables fixed. Yellow: $0 \leqslant \gamma < 0.34$, Red: $0.34 \leqslant \gamma < 0.55$, Blue: $0.55 \leqslant \gamma < 0.75$. Recall all states have the *do nothing* action (self-loops), but are only shown here if contained in one of the optimum policies.

Lastly, recall that DECAY decays the adversaries learning ability the farther into exploring an $\varepsilon-$policy. Further, note that when DECAY $\to 0$ that the state values for states far away from the starting state s_1 became smaller. Due to this, the γ range for the blue optimum policy in Figure 6 changed slightly from the DECAY $= 1$ case as DECAY $\to 0$. For example, when DECAY $= 0.5$, the upper bound for γ in blue optimum policy (see Figure 6) shifted up from 0.75 to 0.76. Additionally, there were changes seen in the optimum policies for γ large enough to push the optimum policy ending state from s_{26} to s_{32}. For instance, when DECAY $= 0.5$ the optimum policy step length for $0.76 < \gamma < 1$ was between 42 and 315. However, the run time for computing Q changed significantly when DECAY was small and γ was large. To illustrate, when DECAY $= 1$ then any run of γ took a maximum of 2 seconds. However, when DECAY $= .5$ and $\gamma = 0.99$ computing Q took 810 seconds.

Experiment 3: Markov Game

A simulation was designed to run the Markov game 500 times. During each run in the simulation, the decay rate γ, the ε for computing the $\varepsilon-$policies, the rate that learning decays DECAY and the maximum policy length before ending the game in a draw were kept fixed at 1, 1, and 150 respectively. However, on every run in the simulation a new set of rewards R_{adv} were assigned to the adversary. During the base case, each adversary action $(s_i, s_j) \in A$ was uniformly assigned a random reward $r_{i,j}$ such that $0 \leqslant r_{i,j} \leqslant r_{max}$ where r_{max} was fixed for the entire simulation.

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

During the operation case, if o was not affecting action (s_i, s_j) then $R(s_i, s_j, o) = r_{i,j}$, otherwise $R(s_i, s_j, o) = r_o$ where $-r_{max} \leqslant r_o \leqslant r_{i,j}$ and was randomly chosen uniformly. Specifically, if the agent was not affecting the edge in G_{adv} that the adversary took, then the agent couldn't affect the rewards the adversary received. Additionally, by setting $-r_{max} \leqslant r_o \leqslant r_{i,j}$ there is an additional assumption imposed that the agents actions either have no effect or are favourable to the agent only. The agent game results for 12 simulations, only changing r_{max} or γ between simulation, are given in Table 2.

r_{max}	γ	WON %	LOST %	DRAW %	TIE %
100	0.2	77.0	9.2	0.0	13.8
100	0.5	97.2	2.8	0.0	0.0
100	0.8	100.0	0.0	0.0	0.0
75	0.2	80.0	9.0	0.0	11.0
75	0.5	98.0	1.8	0.0	0.2
75	0.8	99.6	0.4	0.0	0.0
50	0.2	76.6	12.0	0.0	11.4
50	0.5	97.6	2.4	0.0	0.0
50	0.8	100.0	0.0	0.0	0.0
25	0.2	75.8	11.8	0.0	12.4
25	0.5	95.2	4.4	0.0	0.4
25	0.8	100.0	0.0	0.0	0.0

Table 2: Percentage of agent Markov game outcomes for 12 different simulations, with each simulation running the Markov game 500 times. The r_{max} column shows the max value randomly assigned to the adversary, the minimum value is always zero. The column γ shows the value set for the decay rate γ. All games had $\varepsilon = 1$, DECAY $= 1$, and a max policy length for DRAW set to 150.

Discussion

The Markov decision process shows the optimum policy for the adversary. However, it does not give insight into how an agent should act in response to the adversary's choices. That is where the results of the Markov game are promising, given that the agent won at least 75% of the games and most often greater than 90% of the games, the algorithm used by the agent against the adversary proved to be an effective strategy. Further, the results (seen in Table 2) show that the scale used for the adversary's perceived rewards does not matter. For example, when γ was fixed, the difference between the agent winning percentages was never more that 4.2% between r_{max} values. This is an important result, as results should be consistent no matter the scale used. The results also indicate that the type of adversary (immediate-reward driven or long-term-reward driven) plays a big role in how likely the agent is to win. To illustrate, when $r_{max} = 25$, the agent only won 75% of the time for the immediate-reward driven adversary ($\gamma = 0.2$) v.s. 100% of the time for the more long-term-reward driven adversary ($\gamma = 0.8$).

While these results are exciting, they do assume that the agent is not operating in the environment, i.e., the agent is not trying to traverse the graph or trying to reach some end goal in the same way the adversary is. This assumption makes sense in terms of the gathering intelligence and covertly taking actions to manipulate the adversary. However, for adversary decision-making

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

in combat this Markov game wouldn't work the way it is currently designed. For future works, it would be interesting to test a similar agent strategy when the agent and the adversary are both operating in an environment, each with specific goals and starting points (think war games or capture the flag). Future works should also include repeating all three experiments with different types of graph structures to ensure these results hold up regardless of the specific graph type. Additionally, it is only sometimes realistic to assume that adversary decision-making has the Markov property. Thus, future works could include finding or creating models that match each experiment goal that does not require this property.

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

References

[1] Lorenzo Canese, Gian Cardarilli, Luca Di Nunzio, Rocco Fazzolari, Daniele Giardino, Marco Re, and Sergio Spanò. Multi-agent reinforcement learning: A review of challenges and applications. Journal of Applied Sciences, 4948(11), 2021.

[2] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Designing fast absorbing markov chains. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14, page 849–855. AAAI Press, 2014.

[3] Sherrill Lingel, Matthew Sargent, Timothy R Gulden, Tim McDonald, and Parousia Rockstroh. Leveraging complexity in great-power competition and warfare: Volume 1, an initial exploration of how complex adaptive systems thinking can frame opportunities and challenges. 2021.

[4] Michael Littman. Markov games as a framework for multi-agent reinforcement learning. 2007.

[5] Ismini Lourentzou. Markov games and reinforcement learning. 2007.

[6] Richard Sutton and Andrew Barto. Reinforcement Learning: An Introduction. The MIT Press, Cambridge, MA, USA, 2nd edition, 2020.

[7] William Thompson and James McNeal. Sales planning and control using absorbing markov chains. Journal of Marketing Research, 4(1):62–66, 1967.

[8] Christopher Watkins. Learning from Delayed Rewards. PhD thesis, King's College, Cambridge, UK, May 1989.

[9] Michael Widener. Analysis of soft friend or foe reinforcement learning algorithm in multiagent environment. Master's thesis, Colorado School of Mines, Golden, CO, July 2010.

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

Appendix 1: Algorithms

Algorithm 1: MDP Explore Policy

procedure MDP EXPLOR(G, ε)
  $S \leftarrow$ each state $s_i \in G$
  $A \leftarrow$ each edge $(s_i, s_j) \in G$
  POLICY $\leftarrow s_1$ ▷ s_1 is starting state 1 in G_{adv}
  ABSORB $\leftarrow$ absorbing states of G
  while last state in POLICY not in ABSORB **do**
    s_i = last state in POLICY
    $A' \leftarrow s'_j$ for each $j = 1, .., \text{card}(S)$ s.t. $(s_i, s'_j) \in A$
    $m = \text{argmax}_{s'_j \in A'} Q(s_i, s'_j)$ ▷ breaking ties consistently
    for $s'_j \in A'$ **do**
      if $s'_j = m$ **then**
        $p(s'_j|s_i) = 1 - \varepsilon + \frac{\varepsilon}{\text{card}(A')}$
      else
        $p(s'_j|s_i) = \frac{\varepsilon}{\text{card}(A')}$
    s_j = random choice of $s'_j \in A'$ with probability $p(s'_j|s_i)$
    POLICY $\leftarrow s_j$
  return POLICY

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

Algorithm 2: MDP Optimum Policy

procedure MDP OPTIMUM POLICY(G, R, ε, DECAY, γ)
  $S \leftarrow$ each state $s_i \in G$
  $A \leftarrow$ each edge $(s_i, s_j) \in G$
  ABSORB $\leftarrow$ absorbing states of G
  $Q(s_i, s_j) = 1$ for each $(s_i, s_j) \in A$ ▷ initializing Q

  for forever **do** ▷ do until Q values stop changing
    POLICY $\leftarrow$ MDP EXPLOR(G, ε)
    for forever **do** ▷ do until Q values stop changing
      $\ell = 1$
      for each step s_t in POLICY s.t. s_t not last step in POLICY **do**
        $A' \leftarrow s'_j$ s.t. $(s_{t+1}, s'_j) \in A$
        $m = \max_{s'_j \in A'} Q(s_{t+1}, s'_j)$ ▷ breaking ties consistently
        $Q(s_t, s_{t+1}) \leftarrow Q(s_t, s_{t+1})(1 - \ell) + \ell(r + \gamma \cdot m)$
        $\ell \leftarrow \ell \cdot$ DECAY

  OPTIMUM POLICY $\leftarrow s_1$ ▷ s_1 is starting state 1 in G_{adv}
  SCORE $= 0$
  while last state in OPTIMUM POLICY not in ABSORB **do**
    $s_i =$ last state in OPTIMUM POLICY
    $A' \leftarrow s'_j$ for each $j = 1, .., \text{card}(S)$ s.t. $(s_i, s'_j) \in A$
    $s_j = \text{argmax}_{s'_j \in A'} Q(s_i, s'_j)$ ▷ breaking ties consistently
    if $s_i = s_j$ **then**
      $Q(s_i, s_j) \leftarrow \gamma Q(s_i, s_j)$
    OPTIMUM POLICY $\leftarrow s_j$
    SCORE$\leftarrow Q(s_i, s_j)$
  return OPTIMUM POLICY, SCORE

Algorithm 3: MG Explore Policy

procedure MG EXPLOR(G, ε)
  $S \leftarrow$ each state $s_i \in G$
  $A \leftarrow$ each ADVERSARY action $(s_i, s_j) \in G$
  $O \leftarrow$ an AGENT action o for each $(s_i, s_j) \in A$ ▷ recall $O_i \subset O$ s.t. $(s_i, s'_j) \in A$ for fixed s_i
  set AGT POLICY as empty list
  ADV POLICY $\leftarrow s_1$ ▷ s_1 is starting state 1 in G_{adv}
  ABSORB $\leftarrow$ absorbing states of G

  while last state in ADV POLICY not in ABSORB **do**
    $s_i =$ last state in POLICY
    $A' \leftarrow s'_j$ for each s'_j s.t. $(s_i, s'_j) \in A$
    $min = \text{argmin}_{o' \in O_i} \max_{s'_j \in A'} Q_{adv}(s_i, s'_j, o')$ ▷ breaking ties consistently
    $max = \text{argmax}_{s'_j \in A'} Q_{adv}(s_i, s'_j, o')$ ▷ breaking ties consistently
    for $o' \in O_j$ **do**
      if $o' = min$ **then**
        $p(o'|s_i) = 1 - \varepsilon + \frac{\varepsilon}{\text{card}(O_j)}$
      else
        $p(o'|s_i) = \frac{\varepsilon}{\text{card}(O_j)}$
    for $s'_j \in A'$ **do**
      if $s'_j = max$ **then**
        $p(s'_j|s_i) = 1 - \varepsilon + \frac{\varepsilon}{\text{card}(A')}$
      else
        $p(s'_j|s_i) = \frac{\varepsilon}{\text{card}(A')}$
    $o =$ random choice of $o' \in O_j$ with probability $p(o'|s_i)$
    $s_j =$ random choice of $s'_j \in A'$ with probability $p(s'_j|s_i)$
    AGT POLICY $\leftarrow o$
    ADV POLICY $\leftarrow s_j$
  return AGT POLICY, ADV POLICY

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

Appendix 2: Data

Experiment 1 transition probability data:

$p(s_1|s_1) = 0.3347$
$p(s_1|s_{11}) = 0.02342$
$p(s_1|s_{10}) = 0.12767$
$p(s_1|s_2) = 0.51421$
$p(s_2|s_2) = 0.15662$
$p(s_2|s_3) = 0.15291$
$p(s_2|s_4) = 0.06276$
$p(s_2|s_5) = 0.00881$
$p(s_2|s_6) = 0.51176$
$p(s_2|s_7) = 0.10714$
$p(s_3|s_3) = 0.29679$
$p(s_3|s_8) = 0.65048$
$p(s_3|s_9) = 0.05273$
$p(s_4|s_4) = 0.44875$
$p(s_4|s_8) = 0.08342$
$p(s_4|s_9) = 0.46784$
$p(s_5|s_5) = 0.65189$
$p(s_5|s_8) = 0.05849$
$p(s_5|s_9) = 0.28962$
$p(s_6|s_6) = 0.66921$
$p(s_6|s_8) = 0.30186$
$p(s_6|s_9) = 0.02893$
$p(s_7|s_7) = 0.29742$
$p(s_7|s_8) = 0.42232$
$p(s_7|s_9) = 0.28026$
$p(s_8|s_8) = 1.0$
$p(s_9|s_9) = 0.27426$
$p(s_9|s_{13}) = 0.72574$
$p(s_{10}|s_{10}) = 1.0$
$p(s_{11}|s_{11}) = 0.0651$
$p(s_{11}|s_{12}) = 0.71487$
$p(s_{11}|s_{13}) = 0.22003$
$p(s_{12}|s_{12}) = 1.0$
$p(s_{13}|s_{13}) = 0.85996$
$p(s_{13}|s_{14}) = 0.14004$
$p(s_{14}|s_{14}) = 0.18214$
$p(s_{14}|s_{15}) = 0.1697$
$p(s_{14}|s_{16}) = 0.28159$
$p(s_{14}|s_{17}) = 0.12811$
$p(s_{14}|s_{18}) = 0.18221$
$p(s_{14}|s_{19}) = 0.05625$
$p(s_{15}|s_{15}) = 0.12209$
$p(s_{15}|s_{20}) = 0.87791$
$p(s_{16}|s_{16}) = 0.14303$
$p(s_{16}|s_{20}) = 0.85697$
$p(s_{17}|s_{17}) = 0.52837$
$p(s_{17}|s_{21}) = 0.47163$
$p(s_{18}|s_{18}) = 0.03922$
$p(s_{18}|s_{21}) = 0.96078$
$p(s_{19}|s_{19}) = 0.40899$
$p(s_{19}|s_{21}) = 0.59101$
$p(s_{20}|s_{20}) = 0.65971$
$p(s_{20}|s_{22}) = 0.34029$
$p(s_{21}|s_{21}) = 0.7327$
$p(s_{21}|s_{22}) = 0.2673$
$p(s_{22}|s_{22}) = 0.00136$
$p(s_{22}|s_{23}) = 0.45562$
$p(s_{22}|s_{24}) = 0.18768$
$p(s_{22}|s_{25}) = 0.35534$
$p(s_{23}|s_{23}) = 1.0$
$p(s_{24}|s_{24}) = 0.15598$
$p(s_{24}|s_{27}) = 0.84402$
$p(s_{25}|s_{25}) = 0.26799$
$p(s_{25}|s_{26}) = 0.00438$
$p(s_{25}|s_{27}) = 0.72763$
$p(s_{26}|s_{26}) = 1.0$
$p(s_{27}|s_{27}) = 0.96905$
$p(s_{27}|s_{28}) = 0.03095$
$p(s_{28}|s_{27}) = 0.03653$
$p(s_{28}|s_{28}) = 0.66792$
$p(s_{28}|s_{29}) = 0.29555$
$p(s_{29}|s_{28}) = 0.01254$
$p(s_{29}|s_{29}) = 0.25876$
$p(s_{29}|s_{30}) = 0.25757$
$p(s_{29}|s_{32}) = 0.47113$
$p(s_{30}|s_{29}) = 0.89055$
$p(s_{30}|s_{30}) = 0.09506$
$p(s_{30}|s_{31}) = 0.01439$
$p(s_{31}|s_{30}) = 0.48051$
$p(s_{31}|s_{31}) = 0.51949$
$p(s_{32}|s_{32}) = 1.0$

Experiment 2 action reward data:

$r_{1,1} = 38$
$r_{1,11} = 16$
$r_{1,10} = 57$
$r_{1,2} = 17$
$r_{2,2} = 34$
$r_{2,3} = 40$
$r_{2,4} = 83$
$r_{2,5} = 61$
$r_{2,6} = 0$
$r_{2,7} = 89$
$r_{3,3} = 15$
$r_{3,8} = 100$
$r_{3,9} = 45$
$r_{4,4} = 33$
$r_{4,8} = 100$
$r_{4,9} = 63$
$r_{5,5} = 59$
$r_{5,8} = 100$
$r_{5,9} = 51$
$r_{6,6} = 37$
$r_{6,8} = 100$
$r_{6,9} = 90$
$r_{7,7} = 7$
$r_{7,8} = 100$
$r_{7,9} = 31$
$r_{8,8} = 100$
$r_{9,9} = 6$
$r_{9,13} = 5$
$r_{10,10} = 14$
$r_{11,11} = 18$
$r_{11,12} = 73$
$r_{11,13} = 45$
$r_{12,12} = 13$
$r_{13,13} = 27$
$r_{13,14} = 55$

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

Algorithm 4: MG Optimum Policy

procedure MG OPTIMUM POLICY(G, R_{adv}, ε, DECAY, γ, MAX STEPS)
  $S \leftarrow$ each state $s_i \in G$
  $A \leftarrow$ each ADVERSARY action $(s_i, s_j) \in G$
  $O \leftarrow$ an AGENT action o for each $(s_i, s_j) \in A$ ▷ recall $O_i \subset O$ s.t. $(s_i, s'_j) \in A$ for fixed s_i
  ABSORB $\leftarrow$ absorbing states of G
  $Q_{agt}(s_i, s_j, o) = 1$ for each $(s_i, s_j) \in A$ and $o \in O$
  $Q_{adv}(s_i, s_j, o) = 1$ for each $(s_i, s_j) \in A$ and $o \in O$
  Compute MDP SCORE for ADVERSARY ▷ use algorithm 2

  for forever **do** ▷ do until Q values stop changing
    AGT POLICY, ADV POLICY $\leftarrow$ MG EXPLOR(G, ε)
    for forever **do** ▷ do until Q values stop changing
      $\ell = 1$
      for each step s_t in ADV POLICY s.t. s_t not last step in ADV POLICY **do**
        $A' \leftarrow s'_j$ s.t. $(s_{t+1}, s'_j) \in A$
        $min = \min_{o' \in O_i} \max_{s'_j \in A'} Q_{adv}(s_{t+1}, s'_j, o')$ ▷ breaking ties consistently
        $max = \max_{s'_j \in A'} Q_{adv}(s_{t+1}, s'_j, o')$ ▷ breaking ties consistently
        $Q_{agt}(s_t, s_{t+1}, o) \leftarrow Q_{agt}(s_t, s_{t+1}, o)(1 - \ell) + \ell(-R_{adv}(s_t, s_{t+1}, o) + \gamma \cdot min)$
        $Q_{adv}(s_t, s_{t+1}, o) \leftarrow Q_{adv}(s_t, s_{t+1}, o)(1 - \ell) + \ell(R_{adv}(s_t, s_{t+1}, o) + \gamma \cdot max)$
        $\ell \leftarrow \ell \cdot$ DECAY

  set AGT OPTIMUM POLICY as empty list
  ADV OPTIMUM POLICY $\leftarrow s_1$ ▷ s_1 is starting state 1 in G_{adv}
  MG SCORE $= 0$
  while last state in ADV OPTIMUM POLICY not in ABSORB **do**
    if card(ADV OPTIMUM POLICY) $<$ MAX STEPS **then**
      $s_i =$ last state in ADV OPTIMUM POLICY
      $A' \leftarrow s'_j$ s.t. $(s_i, s'_j) \in A$
      $o = \text{argmin}_{o' \in O_i} \max_{s'_j \in A'} Q_{adv}(s_i, s'_j, o')$
      $s_j = \text{argmax}_{s'_j \in A'} Q(s_i, s'_j, o)$ ▷ breaking ties consistently
      if $s_j = s_i$ **then** for each $o' \in O_i$
        $Q_{adv}(s_i, s_j, o) \leftarrow \gamma Q_{adv}(s_i, s_j, o)$
        $Q_{agt}(s_i, s_j, o) \leftarrow \gamma Q_{adv}(s_i, s_j, o)$
      AGT OPTIMUM POLICY $\leftarrow o$
      ADV OPTIMUM POLICY $\leftarrow s_j$
      MG SCORE$\leftarrow Q_{adv}(s_i, s_j, o)$
    else return DRAW and **break**

  if MG SCORE $<$ MDP SCORE **then return** AGENT WON
  if MG SCORE $>$ MDP SCORE **then return** AGENT LOST
  if MG SCORE $=$ MDP SCORE **then return** TIE

DISTRIBUTION A. Approved for public release; distribution unlimited. Case Number: AFRL-2023-0394 . Dated 20 Apr 2023

$r_{14,14} = 48$
$r_{14,15} = 24$
$r_{14,16} = 11$
$r_{14,17} = 81$
$r_{14,18} = 16$
$r_{14,19} = 89$
$r_{15,15} = 89$
$r_{15,20} = 86$
$r_{16,16} = 19$
$r_{16,20} = 73$
$r_{17,17} = 86$
$r_{17,21} = 78$
$r_{18,18} = 13$
$r_{18,21} = 33$
$r_{19,19} = 46$
$r_{19,21} = 62$
$r_{20,20} = 92$
$r_{20,22} = 87$
$r_{21,21} = 59$
$r_{21,22} = 61$
$r_{22,22} = 78$
$r_{22,23} = 79$
$r_{22,24} = 92$
$r_{22,25} = 98$
$r_{23,23} = 70$
$r_{24,24} = 53$
$r_{24,27} = 4$
$r_{25,25} = 60$
$r_{25,26} = 100$
$r_{25,27} = 1$
$r_{26,26} = 100$
$r_{27,27} = 0$
$r_{27,28} = 26$
$r_{28,27} = 44$
$r_{28,28} = 7$
$r_{28,29} = 16$
$r_{29,28} = 28$
$r_{29,29} = 2$
$r_{29,30} = 27$
$r_{29,32} = 10$
$r_{30,29} = 71$
$r_{30,30} = 26$
$r_{30,31} = 41$
$r_{31,30} = 31$
$r_{31,31} = 6$
$r_{32,32} = 40$