Adversarial Episodic Markov Decision Processes: A review

Mahsa Dalirrooyfard [1]

Abstract

In this review, we examine recent papers about online optimization in adversarial episodic MDPs. In particular, we analyze two papers in detail to learn about optimizing occupancy measures. Furthermore, we explore direct policy optimization and discuss potential future research directions. Lastly, we consider related research fields and how their ideas may be applied to adversarial MDPs.

1. Introduction

In this review paper, we explain recent research and regret bounds on adversarial MDPs with unknown transition function. Mainly, we will focus on two works, (Rosenberg & Mansour, 2019a), where it assumes full information of the loss functions, and (Jin et al., 2020), where works within the more natural and challenging setup of bandit feedback. Both papers use the notion of *occupancy measures* first introduced by (Altman, 1999; Neu et al., 2012) and apply *Online Mirror Descent* (Shalev-Shwartz, 2012) algorithm to tackle, update and optimize these measures. Lastly, we will briefly look at some works such as (Cai et al., 2020) and (Efroni et al., 2020), which try to find the optimized policy directly.

Reinforcement Learning (RL), a branch of machine learning, involves agents learning to behave optimally through repeated interactions with the environment. The purpose of RL is to achieve long-term goals via trial and error, as opposed to supervised learning, which relies on labeled data (Yang & Wang, 2020). By introducing the Markov decision process (MDP) and Q-learning as the solver, (Watkins & Dayan, 1992) laid the foundations of modern RL, based on dynamic programming and TD learning (Klopf & , U.S.).

Typically, the dynamics of the environment are represented as a Markov Decision Process (MDP) with a transition function that is both fixed and unknown. In this context, the papers reviewed consider a general situation where the interaction occurs in episodes with a fixed horizon. In another words, they assume a layered structure state space, where during each episode, the agent initiates their journey from layer 0 and progresses towards the final layer, moving through each layer one by one. The horizon is equal to the number of layers. In each layer, the agent observes its current state, performs an action based on its current policy, experiences the loss associated with that (state, action) pair and moves to the next state in the next layer depending on a transition function unknown to the agent. The ultimate goal of the agent is to minimize its regret, which is defined as the difference between its overall loss (or a convexly measurable function $f_{\mathcal{C}}$ of the loss) and the total loss (respectively $f_{\mathcal{C}}$) incurred by the optimal policy, while striving to achieve a policy that is as close to optimal as possible.

1.1. Motivation

BGP (Border Gateway Protocol) routing is considered as a motivation in (Rosenberg & Mansour, 2019a) for adversarial MDPs. In this context, adversarial MDP can be used to improve the decision-making process of the BGP routers.

An autonomous system (AS) on the internet uses the BGP protocol to exchange routing information between routers. BGP routers make routing decisions based on the information received from other routers and the policies defined by the network administrator. Other routers on the internet, however, may behave in unpredictable ways and may not always follow the expected norms, resulting in sub-optimal routing decisions.

Adversarial MDP can help improve the decision-making process of BGP routers by modeling the behavior of other routers as an adversarial environment. BGP routers learn to adapt to the behavior of other routers and make optimal decisions in the face of unexpected or adversarial behavior.

The main application of adversarial MDP is in multi-agent reinforcement learning, where agents are instructed to optimize their policies in competitive or cooperative environments.

Some other applications and motivations for adversarial MDPs include but not limited to:

[1]Department of Mathematics, London School of Economics and Political Science, London, United Kingdom. Correspondence to: Mahsa Dalirrooyfard <m.dalirrooy-fard@lse.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

- **Security**
  In security scenarios, such as cyber-attackers or physical security, adversarial MDPs can be used to model attacker behavior as the adversarial environment and the defenders as the learning agents. Agents can learn to optimize their policies so that they can exploit or defend against different attack strategies.

- **Robotics**
  Another application for adversarial MDP is using them to model the interaction between robots in a field, competing with each other to acheive their own goal.

- **Game Theory**
  One can view a game and the interactions between the players as an adversarial MDP. Then the optimal policy of the agent would be equivalent to the player's strategy in the equilibrium of the game.

1.2. Related work

Markov decision processes are the most widely used model in reinforcement learning. First introduced in (Puterman, 1994), the model assumes that the loss/reward functions and the environment dynamics do not change over time; However that is not true in the real world.
In order to tackle this problem and propose more realistic models, adversarial MDP models were presented by (Even-Dar et al., 2009) and (Yu et al., 2009). In adversarial MDPs, the loss functions can change arbitrarily. These two initial papers consider the setting with known transition function and full information feedback on loss functions and they both use a continuous model rather than the episodic MDPs studied in this paper.
(Neu et al., 2010b) considers episodic MDP with known transition function and bandit feedback where the learner only observes the losses associated with its actions.
The first work considering unknown transition function was (Neu et al., 2012) proposing Perturbed Optimistic Policy algorithm. Later (Rosenberg & Mansour, 2019a) improves their regret bound in unknown transition function and full information feedback setting by combining ideas from Online Mirror Descent algorithm (Shalev-Shwartz, 2012) and UCRL-2 algorithm (Auer et al.), which we will investigate in detail later. Shortly afterward, the same authors (Rosenberg & Mansour, 2019b) propose the first algorithm for unknown transition function and bandit feedback setting. Their algorithm achieves $\tilde{O}(T^{3/4})$ or $\tilde{O}(\sqrt{T}/\alpha)$ regret bound where T is the number of episodes and it is assumed that all states are reachable with positive small probability α. Their regret bound then is improved by (Jin et al., 2020) to $\tilde{O}(\sqrt{T})$ removing the assumption of minimum probability for state reachability.
All of these algorithms agree with the lower bound $\Omega(L\sqrt{|X||A|T})$ for regret, proved by (Jin et al.) where L is the length of each episode, $|X|$ the number of states, $|A|$ the number of Actions and T the number of episodes as before.
Instead of optimizing occupancy measures as in (Zimin & Neu, 2013; Rosenberg & Mansour, 2019a; Jin et al., 2020), one can find the best policy directly. (Cai et al., 2020) and (Efroni et al., 2020) proved $\tilde{O}(\sqrt{K})$ and $\tilde{O}(\sqrt{S^2AH^4K})$ regret bounds for full information and bandit feedback settings respectively, where S is the size of state space, A is the size of action space, H is the horizon of the MDP (length of each episode) and K being the total number of episodes. The second bound is equivalent to the one in (Rosenberg & Mansour, 2019a) and matches the upper bound of $\tilde{O}(\sqrt{K^{2/3}})$ by (Neu et al., 2010a).
Later (Luo et al., 2021) improved the regret bound of (Efroni et al., 2020) from almost $\tilde{O}(\sqrt{K^{2/3}})$ to $\tilde{O}(\sqrt{K})$.
There are several other settings that adversarial MDPs have been studied, such as delayed feedback (Rosenberg et al., 2022) or as a stochastic problem assuming the transition and loss functions change with time (Adversarial transition function) (Cheung et al., 2019; Lykouris et al., 2021). The latter paper proposes a regret bound for when a constant number of episodes are being corrupted.
Another line of works are linear MDPs, where the loss/reward and transition functions are assumed to be linear. Starting with (Cai et al., 2020), they demonstrated a regret bound of $\tilde{O}(\sqrt{K})$, K being the number of episodes, while considering the full information feedback setting. Then, (Neu & Olkhovskaya, 2021) proved the same bound for bandit feedback setup. Later, (Luo et al., 2021) showed $\tilde{O}(K^{14/15})$ regret bound in the more general setting of the unknown dynamics and bandit feedback. Their bound was improved to $\tilde{O}(K^{6/7})$ in the most recent work (Sherman et al., 2023).

Table 1 shows a summary of the most important algorithms for this review along with their setting and more exact regret bound.

2. Problem Formulation

A tuple in the format $M = (X, A, P, \{\ell_t\}_{t=1}^T)$ defines an adversarial Markov decision process. X denotes the finite state space, while A indicates the finite action space. $P : X \times A \times X \to [0, 1]$ is the transition function, where $P(x'|x, a)$ denotes the probability of transferring to state x' by executing action a in state x. $\{\ell_t\}_{t=1}^T$ is the sequence of loss functions with slightly different definitions in the main papers of this review:

- In (Rosenberg & Mansour, 2019a), ℓ_t is a function from $X \times A \times X$ to $[0, 1]^d$ with no statistical assumption and they can be chosen arbitrarily. (Here d is an arbitrary positive integer).

Table 1. Comparison of the regret bounds for some recent adversarial MDP algorithms. Third column shows whether the model assumes an unknown transition function, column four differs between the setting with full information on the loss function or the bandit feedback setup, column five indicates whether the algorithm uses occupancy measures or not, column six shows if the function estimation is linear or not and in the regret bound column, L is the length or horizon of each episode, $|X|$ is the size of state space, $|A|$ is the size of action space, T is the total number of episodes and d is the feature dimension and equal to $|X|^2|A|$.

ALGORITHM	PAPER	UNKNOWN TRANSITION	BANDIT FEEDBACK	OCCUPANCY MEASURE	LINEAR	REGRET BOUND
UC-O-REPS	(ROSENBERG & MANSOUR, 2019A)	√	×	√	×	$\tilde{O}(L\|X\|\sqrt{\|A\|T})$
SHIFTED BANDIT UC-O-REPS	(ROSENBERG & MANSOUR, 2019B)	√	√	√	×	$\tilde{O}(L^{3/2}\|X\|\|A\|^{1/4}T^{3/4})$
UOB-REPS	(JIN ET AL., 2020)	√	√	√	×	$\tilde{O}(L\|X\|\sqrt{\|A\|T})$
POMD	(EFRONI ET AL., 2020)	√	√	×	×	$\tilde{O}(L\|X\|\sqrt{\|A\|}T^{2/3})$
OPPO	(CAI ET AL., 2020)	√	×	×	√	$\tilde{O}(\sqrt{d^2L^4T})$

- In (Jin et al., 2020), They consider a one dimension loss function ($d = 1$). Here, ℓ_t is a function from $X \times A \to [0, 1]$.

The definitions are easily comparable and convertible. Moreover, both papers assume the state space X can be divided into $L + 1$ non-intersecting layers, namely $X_0, X_1, ..., X_L$. X_0 and X_L are singletons and Only transitions between adjacent layers are permitted (loop-free). These similar assumptions with the goal to simplify the notation, the arguments and analysis, can also be seen in previous works such as (Neu et al., 2012) and (Zimin & Neu, 2013). It is worth reiterating that the transition function is not known to the learning agent in the primary papers reviewed.
At the begining of the process, the environment (advisory) chooses an MDP and lets the agent know only about the state space, its layer structure and the action space. Then the learning process happens in T rounds or episodes. In each episode, the agent starts from the state X_0, chooses actions based on its current policy and moves forward layer by layer until it reaches X_L. In (Rosenberg & Mansour, 2019a), the agent sees the whole loss function of that episode when the episode ends, while in (Jin et al., 2020), the agent only observes the loss amount (bandit feedback) for the visited state-action pair (x_t, a_t) instantly after its move. In all the adversarial MDP problems, the ultimate goal of the agent is to find the best stochastic policy. The policy is simply a function $\pi : X \times A \to [0, 1]$ where $\pi(a|x)$ is the probability that the agent chooses action a in state x.
Before exploring the methods and algorithms of the papers in detail, there a few common concepts and ideas which are also useful for future work.

2.1. Occupancy Measures

To reformulate the objective as an online convex optimization problem, the papers use the concept of occupancy measures (Altman, 1999; Neu et al., 2012; Zimin & Neu, 2013).

For a stochastic policy π and a transition function P, the occupancy measure $q^{P,\pi}$ is defined as follows on $X \times A \times X$:

$$q^{P,\pi}(x, a, x') = Pr[x_k = x, a_k = a, x_{k+1} = x'|P, \pi]$$

Where x belongs to layer k of the state space meaning X_k and x' belongs to X_{k+1}.
Any occupancy measure $q^{P,\pi}$ has a few basic properties (Rosenberg & Mansour, 2019a; Jin et al., 2020) that are worth noting:

1. Since $q^{P,\pi}$ is a probability function, and the agent moves layer by layer, or in another words the MDP is loop free, in every episode, each layer is visited exactly once. Therefore, for every layer $k \in \{0, 1, 2, ..., L-1\}$ we have:

$$\sum_{x \in X_k} \sum_{a \in A} \sum_{x' \in X_{k+1}} q(x, a, x') = 1 \tag{1}$$

2. With the exception of states X_0 and X_L, the probability of transitioning to a state from the previous layer is equivalent to the probability of moving from that same state to the next layer. Therefore, for every layer $k \in \{1, 2, ..., L-1\}$ and a state x in layer k:

$$\sum_{x' \in X_{k-1}} \sum_{a \in A} q(x', a, x) = \sum_{x' \in X_{k+1}} \sum_{a \in A} q(x, a, x') \tag{2}$$

3. From any occupancy measure, we can extract a transition function and a policy as follows:

$$P^q(x'|x, a) = \frac{q(x, a, x')}{\sum_{y \in X_{Layer(x)+1}} q(x, a, y)} \tag{3}$$

$$\pi^q(a|x) = \frac{\sum_{x' \in X_{Layer(x)+1}} q(x, a, x')}{\sum_{b \in A} \sum_{x' \in X_{Layer(x)+1}} q(x, b, x')} \tag{4}$$

Then, instead of finding the best policy directly, The primary reviewed papers use the online mirror descent (OMD) method to choose an optimized occupancy measure q_t for each episode $t \in \{1, 2, ..., T\}$, and so the agent moves along the layers of the episode t based on the policy derived from q_t.

2.2. Confidence Sets

Recall that (Jin et al., 2020) works with bandit feedback which is a harder setup but also closer to real world applications comparing to the full information setting in (Rosenberg & Mansour, 2019a). Another important difference in (Jin et al., 2020), is the tighter confidence sets for the transition function and as a result, lower bias for the loss function estimators.
Since the agent is not aware of the transition function, both papers use the same approach to calculate an empirical transition function estimator. The algorithms work in epochs. Each epoch consists of a number of consecutive episodes. $N_i(x, a)$ and $M_i(x, a, x')$ are respectively the number of times the (state, action) pair (x, a) and the (state, action, next state) triple (x, a, x') are visited before epoch i. Then the empirical transition function is calculated as follows for each epoch i:

$$\bar{P}_i(x'|x, a) = \frac{M_i(x, a, x')}{max\{1, N_i(x, a)\}}$$

The confidence set $\Delta(M, i)$ for epoch i is the set of all occupancy measures q such that their derived transition function P^q is within a defined distance from the empirical transition function $\bar{P}_i$. More specifically:

- In (Rosenberg & Mansour, 2019a),

  $$|P^q(.|x, a) - \bar{P}_i(.|x, a)| \leq \epsilon_i(x, a)$$

  for every pair $(x, a) \in X \times A$ where

  $$\epsilon_i(x, a) = \sqrt{\frac{2|X_{Layer(x)+1}| \ln \frac{T|X||A|}{\delta}}{max\{1, N_i(x, a)\}}}$$

  and

- In (Jin et al., 2020),

  $$|P^q(x'|x, a) - \bar{P}_i(x'|x, a)| \leq \epsilon'_i(x'|x, a)$$

  for every triple $(x, a, x') \in X_k \times A \times X_{k+1}$ for $k = 0, 1, 2, ..., L-1$ where

  $$\epsilon'_i(x'|x, a) = 2\sqrt{\frac{\bar{P}_i(x'|x, a) \ln \frac{T|X||A|}{\delta}}{max\{1, N_i(x, a) - 1\}}} + \frac{14 \ln \frac{T|X||A|}{\delta}}{3max\{1, N_i(x, a) - 1\}}$$

In ϵ_i and ϵ'_i functions, δ is a confidence parameter in $[0, 1]$. Both papers use the calculated confidence set as their estimation for the unknown transition function.

2.3. Online Mirror Descent

Both papers use the same OMD component in their algorithms in order to choose the occupancy measure of episode $t + 1$. The goal is to minimize a function of loss/loss estimator, as well as staying close to the occupancy measure of episode t. For the statistical distance between two occupancy measure, they use unnormalized KL divergence (Kullback & Leibler, 1951; Kullback, 1959).
Mirror descent is an iterative optimization algorithm used in mathematics and machine learning to find a local minimum of a differentiable function. This algorithm generalizes other optimization techniques like gradient descent and multiplicative weights. The algorithm involves minimizing an approximation of the objective function in addition to a proximity term, which is expressed using the KL-divergence (Or more generally, Bregman divergence) between the previous and updated solution estimates.
The Mirror Descent (MD) method was initially introduced in the batch setting by (Nemirovski & Yudin, 1983). However, it was only in 1997 that the mirror descent scheme was used for the first time in the online setting, by (Kivinen & Warmuth, 1997).

2.4. Inverse Importance-Weighted Estimators

Although the full information setting in (Rosenberg & Mansour, 2019a), gives the agent the full loss function after each episode, in the bandit feedback setting, the algorithm needs to estimate the loss functions from the observations it has.

Inverse Importance Weighting (IIW) is a widely acceptable statistical technique to estimate a population or function when one only has partial information or a sample, not representing the whole population.

The IIW estimator considers a new weight for each sample which is the inverse of its probability of being included in the sample. The weights compensate for the bias made by choosing non random samples and produces an unbiased estimator in the end.

Here, to estimate the loss function, one can use the inverse of the occupancy measure for each (state, action) pair (x, a) that was visited during traversing the layers of the state space in some episode t, which gives an unbiased estimator. But the problem, also mentioned in (Jin et al., 2020), is that to use $q^{P,\pi_t}(x, a)$ one needs to know the transition function P. To address this issue, (Jin et al., 2020), finds the maximum occupancy measure over all transition functions in the confidence set calculated for the pair (x, a) (See section 2.2). Assuming $\mathcal{P}_i$ to be the confidence set for

the epoch i where episode t belongs to, then the *upper occupancy bound* is defined as follows:

$$u_t(x,a) = \max_{\hat{P} \in \mathcal{P}_i} q^{\hat{P},\pi_t}(x,a)$$

The upper occupancy bound shows the maximum probability of visiting (x,a) among all possible transition functions.

To obtain a high probability regret bound, (Jin et al., 2020) embraces the concept of *implicit exploration* as proposed by (Neu, 2015) and adds an exploration parameter γ to the upper occupancy bound to get the final estimator for $q^{P,\pi_t}(x,a)$. Therefore, The loss function estimator used in (Jin et al., 2020) UOB-REPS algorithm is as follows:

$$\hat{\ell}_t(x,a) = \frac{\ell_t(x,a)}{u_t(x,a)+\gamma}\mathbb{I}\{x_{k(x)} = x, a_{k(x)} = a\}$$

where $k(x)$ is the index of the layer of state x. Note that $\hat{\ell}_t(x,a)$ is a biased estimator but it is shown in (Jin et al., 2020) that the bias introduced by this approach is reasonably small.

3. Algorithms

In this section, we will further explain the algorithms **UC-O-REPS** (Rosenberg & Mansour, 2019a) and **UOB-REPS** (Jin et al., 2020) and their regret bounds.

3.1. UC-O-REPS

This algorithm receives the state space X, the action space A, number of episodes T, the performance criteria function $f_{\mathcal{C}}$, the optimization parameter η (related to the OMD method) and the confidence parameter δ as inputs.
It starts by initializing the policy π_1 and the occupancy measure q_1, by putting equal probability of $\pi_1(a|x) = \frac{1}{|A|}$ for each $a \in A, x \in X$ and also $q_1(x,a,x') = \frac{1}{|X_k||A||X_{k+1}|}$ for each $k \in \{0,1,2,...,L-1\}$ and $(x,a,x') \in X_k \times A \times X_{k+1}$.
The algorithm works in epochs where each one consists of a number of consecutive episodes. In each episode, the agent moves layer by layer in the state space based on its current policy π_t and at the end, receives the loss function from the environment. Moreover, it counts the number of times visiting each pair of (state, action) and each triple of (current state, action, next state). If the number of visiting a pair is doubled from the start of the epoch, then it moves to the next epoch.
At the end of each episode, it uses the OMD optimization techniques to find the optimal occupancy measure and policy for the next episode.

3.2. UOB-REPS

The inputs for this algorithm are similar to the one above, except that instead of $f_{\mathcal{C}}$, it receives an *exploration parameter* γ which is used in the loss function estimators.
The initialization for the policy and the occupancy measure is exactly the same as UC-O-REPS. UOB-REPS also works in epochs and the epoch increments happens the same way. After traversing the state space based on the current policy, the algorithm needs to estimate the loss function, since despite UC-O-REPS, here the agent only receives the loss value for pairs (state, action) that it has visited and not the entire loss function.
UOB-REPS then uses the techniques explained in section 2.4 to estimate the loss function (The name for the algorithm, *Upper Occupancy Bound Relative Entropy Policy Search*, also comes from their estimation method for the loss function). Lastly, with the similar OMD optimization component, the algorithm calculates the optimal policy and occupancy measure for the next episode.

3.3. Regret Analysis

The goal of the learning agent is to minimize the regret; That is the difference between L_T gained by the algorithm and the one by the best fixed policy in hindsight where L_T can be a convexly measurable performance function over the losses. (Rosenberg & Mansour, 2019a) works with this general definition of a convexly measurable performance function $f_{\mathcal{C}}$ while (Jin et al., 2020) considers a specific example of $f_{\mathcal{C}}$, the total loss of the learner. More specifically:

$$L_T(\pi) = \sum_{t=1}^{T} \mathbb{E}\left[\sum_{k=0}^{L-1} \ell_t(x_k,a_k)|P,\pi\right]$$

is the total loss for a fixed policy π,

$$L_T = \sum_{t=1}^{T} \mathbb{E}\left[\sum_{k=0}^{L-1} \ell_t(x_k,a_k)|P,\pi_t\right]$$

is the total loss of the learner and finally:

$$R_T = L_T - \min_{\pi} L_T(\pi)$$

is the regret which the agent wants to minimize over all possible stochastic policies.

For the general convexly measurable performance function $f_{\mathcal{C}}$, let $R_T^{\mathcal{C}}$ be the regret. The following theorem is the main regret bound result of (Rosenberg & Mansour, 2019a).

Theorem 3.1. *Consider an episodic loop-free adversarial MDP $M = (X, A, P, \{\ell_t\}_{t=1}^T)$. Let $f_{\mathcal{C}}$ be a convexly measurable performance function that is F-Lipschitz. Let the optimization parameter be $\eta = \sqrt{\frac{\ln \frac{|X|^2|A|}{\delta}}{\frac{L^2}{F^2 T}}}$. Then with probability at least $1-2\delta$, the algorithm UC-O-REPS achieves*

the following regret bound:

$$R_T^C \leq 15FL|X|\sqrt{T|A|\ln\frac{T|X||A|}{\delta}}$$

If we let the convexly measurable performance function to be the total loss of the learner, since this function is 1-Lipschitz and by putting $\delta = \frac{|X||A|}{T}$, the regret bound will get simplified to

$$R_T^{TL} \leq 25L|X|\sqrt{T|A|\ln T}$$

For the total loss of the learner as the performance function, the following is the main regret bound result of (Jin et al., 2020).

Theorem 3.2. *Let M be the MDP similar to above. Let the optimization and exploration parameters to be $\eta = \gamma = \sqrt{\frac{L\ln(L|X||A|/\delta)}{T|X||A|}}$. Then with probability at least $1-9\delta$, the algorithm UOB-REPS achieves the following regret bound:*

$$R_T = O\left(L|X|\sqrt{T|A|\ln\frac{T|X||A|}{\delta}}\right)$$

(Rosenberg & Mansour, 2019a) proves the regret bound by partitioning it into two parts, the error coming from the transition function estimation and the error that comes from choosing sub-optimal occupancy measure and policy. (Jin et al., 2020) divides the regret into four terms, the error of occupancy measure estimation, the regret introduced by the OMD component and two bias terms for the loss estimators. Then, both papers prove upper bounds for each part. We omit the details here.

The regret bounds are similar, first achieved for the full information setting in (Rosenberg & Mansour, 2019a) and then for the harder and more natural bandit feedback setup by (Jin et al., 2020).

4. Policy Optimization Methods

While (Rosenberg & Mansour, 2019a) and (Jin et al., 2020) use the notion of occupancy measure and OMD method to tackle with the adversarial MDP problem, (Cai et al., 2020) and (Efroni et al., 2020) propose optimistic policy optimization algorithms.

Reinforcement Learning (RL) commonly utilizes Policy Optimization (PO) as one of its most prevalent methods. These methods work by incrementally modifying the policy itself, making PO algorithms highly versatile and applicable to a diverse range of RL tasks. Notable examples of PO algorithms include policy-gradient algorithms (Sutton et al., 1999), natural policy gradient (Kakade, 2001) and trust region policy optimization (TRPO) (Schulman et al., 2015).

Other than policy gradient method, there is another way also to optimize a policy which is the derivative-free optimization (DFO) algorithms. These easy to implement algorithms change the policy parameters by small amounts in different ways, observe the performance and move towards a good performance. Not being able to scale well with the number of parameters is the drawback of DFO algorithms.

(Cai et al., 2020) considers the adversarial environment with unknown transition function but full information on the loss functions, a similar setting to (Rosenberg & Mansour, 2019a), and introduces Optimistic Proximal Policy Optimization (OPPO) algorithm. The algorithm has two phases, policy improvement and policy evaluation. In policy improvement step, they insure to improve the expected total reward of the agent or reduce the expected total loss while staying close to the previous policy based on KL-divergence distance measurement. The paper proves $\tilde{O}(\sqrt{d^2L^4T})$ regret bound where T is the number of episodes, L is the length of each episode (the number of the layers of the state space) and $d = |X|^2|A|$ where X and A are the state and action space respectively.

(Efroni et al., 2020) introduces the Policy Optimization by Mirror Descent (POMD) algorithm, once for the stochastic environment and then for adversarial, both with bandit feedback and unknown transition function. Similar to OPPO, this algorithm also alternates between two phases of policy improvement and evaluation and makes sure the new policy remains close to the previous one by the same KL-divergence distance. In the stochastic setting, POMD has the same regret bound as (Rosenberg & Mansour, 2019a) and in adversarial setup, the bound is slightly bigger in terms of T, the number of episodes, see table 1.

In addition to different main method used to solve the adversarial MDP, (Rosenberg & Mansour, 2019a) and (Jin et al., 2020) have a number of other model and technique differences with (Cai et al., 2020) and (Efroni et al., 2020) which are worth noticing:

- **Transition Function Model**
  To tackle with the issue of unknown transition function in the MDP, (Rosenberg & Mansour, 2019a) and (Jin et al., 2020) make a confidence interval including all the possible transition functions that are close enough to the empirical transition function calculated from the agent traverse in the state space during an episode. However, (Cai et al., 2020) and (Efroni et al., 2020), both add a bonus term, a function of the (state, action) pair, to compensate on the unknown transition.
- **Loss Estimation**
  In the bandit feedback setting, agent only observes the loss for the (state, action) pairs that it has visited. To estimate the entire loss function, (Jin et al., 2020)

uses Inverse Importance Weighting method explained in section 2.4 while (Efroni et al., 2020) similar to its transition function estimation, uses a bonus term depending on the (state, action) pair.

- **Optimization Problem Solution**
(Rosenberg & Mansour, 2019a) and (Jin et al., 2020) tackle the optimization problem across the complete state-action domain, whereas (Cai et al., 2020) and (Efroni et al., 2020) employ a closed-form solution on a per-state basis.

In spite of the success of neural networks for policy optimization in achieving state-of-the-art results in reinforcement learning across a variety of domains, theoretical understanding of policy optimization's computational and sample efficiency is limited to linear function approximations coupled with finite-dimensional feature representations. Even with infinite data, policy optimization was uncertain until recently from a computational standpoint whether it would converge to the globally optimal policy in a finite number of iterations. Statistically, it is also unclear how to achieve a globally optimal policy with limited regret or sample complexity (Cai et al., 2020). This lack of understanding impedes the development of principled, effective, and efficient algorithms.

5. Discussion

We have studied two significant methods for addressing adversarial MDP issues: one involves optimizing occupancy measures, while the other focuses on improving the policy directly. It is also possible to merge the ideas presented in the papers discussed in this review. This can be accomplished by utilizing bonus functions from the works of (Cai et al., 2020; Efroni et al., 2020) to recompense for errors in loss and transition function estimation, while also updating occupancy measures as suggested by (Rosenberg & Mansour, 2019a; Jin et al., 2020).

Up to this point, works on MDP problems have primarily focused on environments with either known or unknown transition functions and full-information or bandit-loss feedback. Limited research has been conducted on environments with unrestricted delayed feedback, as seen in (Rosenberg et al., 2022). Another area of potential interest would be an environment in which the agent not only receives bandit feedback but also experiences a delay.

Sometimes in the real world, in case the environment communicates through a network system with the agent, the system may experience glitches that prevent the agent from receiving feedback, or the feedback delay may be so long that it is effectively lost. In such flawed settings, it can be challenging but intriguing to see how an agent can effectively learn.

Another possible flawed system in real applications is that the feedback received is within a confidence interval of the real loss value of that (state, action) pair and cannot be determined exactly. It is interesting to see which of the algorithms so far can be generalized to adopt and work efficiently in this setting.

In addition to delay in observing the loss/reward for, after performing an action a, an agent might also need to make its next decision rapidly while not knowing which state they are in as a result of a. Therefore, they need to decide on a probability distribution on the possible next actions based on the possible states they might be in.

A way to approach MDP problems is to consider them as games and apply game theory techniques to solve them. It is possible to view this game's equilibrium as the optimal solution for the agent. A perfect information game can be seen as equivalent to a known environment in MDP, while an unknown transition or loss function can be represented as an incomplete or imperfect information game.

The loss/reward in an MDP can be considered as the player's payoff in a game. An adversarial MDP, where the agent's loss is equal to the environment's gain, can be viewed as a zero-sum game, first introduced by John Von Neumann and Oscar Morgenstern. In such games, Von Neumann's *Minimax Theorem* (Von Neumann, 1928) can be applied, which states that there are strategies that each player should use that will result in the best outcome for both players. In other words, the best strategy for the agent player would be its optimal solution to the adversarial MDP.

Repeated games and reinforcement learning algorithms have been sharing ideas for some time now. Both address similar challenges such as coordination, learning from loss/reward full or bandit feedback and strategic or adversarial interactions. As a result, insights and techniques from one field can often be applied to the other.

Examples include but not limited to Markov games (Van Der Wal, 1981; Littman, 1994), Stochastic games (See (Wei et al., 2017) as an example) and multi-agent RL which corresponds to the learning problem in a multi-agent system in which multiple agents learn simultaneously in a shared environment, each trying to maximize their reward.

Moreover, many famous games have been solved by RL algorithms such as Go (Silver, 2016) or Chess (Silver et al., 2018).

Other than the applications of adversarial MDPs described above and in 1.1, this section concludes with several intriguing specific uses of RL and adversarial MDPs. These can inspire future research or enhancements:

- **Resource Allocation Management**
  The paper (Mao et al., 2016) explains and designs an algorithm such that an agent learns how to allocate available computer resources to existing jobs while minimizing average deceleration of jobs. The state space would be the current resource allocation along with other resource needs of current jobs and the action space consists of assigning different resources to different jobs. The design of jobs' resource requirements and their arrival can be assumed to be done by an adversary.

- **Traffic Light Control**
  Although not yet used in real world, (Arel et al., 2010) by using multi agent RL, design a traffic light controller to reduce the congestion. The state space would be a vector representing the traffic flow of the related streets and action space consists of the different phases of the traffic light for all the streets.

- **Bidding and Advertising**
  Bidding platforms can be modeled as adversarial MDPs where the state space is the current revenue of different agents while the action space is their bid. Here the gain of an agent is the loss of the other, which makes the model adversarial. One recent paper in this area is the Alibaba group research (Song et al., 2018).

6. Conclusion

In this review paper, we considered the problem of adversarial MDPs with unknown transition function and explored two possible techniques to tackle with the challenges of this problem; One by updating occupancy measures in each episode and deriving the optimal policy from them. This approach was taken by (Rosenberg & Mansour, 2019a) and (Jin et al., 2020) in full information and bandit feedback setting respectively, each proved the regret bound $\tilde{O}(L|X|\sqrt{|A|T})$ for its own setup.

The second technique was to optimize the policy directly where each episode consisted of two steps, policy improvement and policy evaluation. We explored the papers (Cai et al., 2020; Efroni et al., 2020) for the latter method. (Cai et al., 2020) proved the bound $\tilde{O}(\sqrt{d^2L^4T})$ for the full information on the loss function and assuming linear function approximation, while (Efroni et al., 2020) established the bound $\tilde{O}(L|X|\sqrt{|A|T^{2/3}}))$ in the bandit feedback setup.

References

Altman, E. Constrained markov decision processes. 1999.

Arel, I., Liu, C., Urbanik, T., and Kohls, A. Reinforcement learning-based multi-agent system for network traffic signal control. *Intelligent Transport Systems, IET*, 4:128 – 135, 07 2010.

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. Provably efficient exploration in policy optimization. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1283–1294. PMLR, 13–18 Jul 2020.

Cheung, W.-C., Simchi-levi, D., and Zhu, R. Reinforcement learning under drift. 06 2019.

Efroni, Y., Shani, L., Rosenberg, A., and Mannor, S. Optimistic policy optimization with bandit feedback. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Even-Dar, E., Kakade, S. M., and Mansour, Y. Online markov decision processes. 2009. ISSN 0364765X, 15265471.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Jin, C., Jin, T., Luo, H., Sra, S., and Yu, T. Learning adversarial Markov decision processes with bandit feedback and unknown transition. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4860–4869. PMLR, 13–18 Jul 2020.

Kakade, S. M. A natural policy gradient. In Dietterich, T., Becker, S., and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

Kivinen, J. and Warmuth, M. K. Relative loss bounds for multidimensional regression problems. In *Proceedings of the 10th International Conference on Neural Information Processing Systems*, NIPS'97, pp. 287–293, Cambridge, MA, USA, 1997. MIT Press.

Klopf, A. and (U.S.), A. F. C. R. L. *Brain Function and Adaptive Systems: A Heterostatic Theory*. Special reports. Air Force Cambridge Research Laboratories, Air Force Systems Command, United States Air Force, 1972.

Kullback, S. *Information Theory and Statistics*. Wiley, 1959.

Kullback, S. and Leibler, R. A. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951. doi: 10.1214/aoms/1177729694.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, ICML'94, pp. 157–163, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1558603352.

Luo, H., Wei, C.-Y., and Lee, C.-W. Policy optimization in adversarial mdps: Improved exploration via dilated bonuses. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, pp. 22931–22942. Curran Associates, Inc., 2021.

Lykouris, T., Simchowitz, M., Slivkins, A., and Sun, W. Corruption-robust exploration in episodic reinforcement learning. In Belkin, M. and Kpotufe, S. (eds.), *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pp. 3242–3245. PMLR, 15–19 Aug 2021.

Mao, H., Alizadeh, M., Menache, I., and Kandula, S. Resource management with deep reinforcement learning. HotNets '16, pp. 50–56, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450346610.

Nemirovski, A. and Yudin, D. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983.

Neu, G. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *NIPS*, 2015.

Neu, G. and Olkhovskaya, J. Online learning in mdps with linear function approximation and bandit feedback. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 10407–10417. Curran Associates, Inc., 2021.

Neu, G., Antos, A., György, A., and Szepesvári, C. Online markov decision processes under bandit feedback. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010a.

Neu, G., György, A., and Szepesvári, C. The online loop-free stochastic shortest-path problem. pp. 231–243, 01 2010b.

Neu, G., Gyorgy, A., and Szepesvari, C. The adversarial stochastic shortest path problem with unknown transition probabilities. In Lawrence, N. D. and Girolami, M. (eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 805–813, La Palma, Canary Islands, 21–23 Apr 2012. PMLR.

Puterman, M. L. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.

Rosenberg, A. and Mansour, Y. Online convex optimization in adversarial Markov decision processes. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5478–5486. PMLR, 09–15 Jun 2019a.

Rosenberg, A. and Mansour, Y. Online stochastic shortest path with bandit feedback and unknown transition function. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019b.

Rosenberg, A., tal lancewicki, and Mansour, Y. Learning adversarial markov decision processes with delayed feedback. 2022.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Sherman, U., Koren, T., and Mansour, Y. Improved regret for efficient online reinforcement learning with linear function approximation, 2023.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362 (6419):1140–1144, 2018.

Silver, D., H.-A. M. C. e. a. Mastering the game of go with deep neural networks and tree search. volume 529, pp. 484–489, 2016.

Song, C., Jin, J., Li, H., Gai, K., Wang, J., and Zhang, W. Real-time bidding with multi-agent reinforcement learning in display advertising. 2018.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Solla, S., Leen, T., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

Van Der Wal, J. Stochastic dynamic programming. Mathematical Centre Tracts, Morgan Kaufmann, 1981.

Von Neumann, J. Zur theorie der gesellschaftsspiele. volume 100, pp. 295–320. Math. Ann, 1928.

Watkins, C. J. and Dayan, P. Q-learning. volume 8, pp. 279–292, 1992.

Wei, C.-Y., Hong, Y.-T., and Lu, C.-J. Online reinforcement learning in stochastic games. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yang, Y. and Wang, J. An overview of multi-agent reinforcement learning from game theoretical perspective. 2020.

Yu, J. Y., Mannor, S., and Shimkin, N. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

Zimin, A. and Neu, G. Online learning in episodic markovian decision processes by relative entropy policy search. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2013.