Reinforcement Learning for Constrained Markov Decision Processes

Ather Gattami
AI Sweden

Qinbo Bai
Purdue University

Vaneet Aggarwal
Purdue University

Abstract

In this paper, we consider the problem of optimization and learning for constrained and multi-objective Markov decision processes, for both discounted rewards and expected average rewards. We formulate the problems as zero-sum games where one player (the agent) solves a Markov decision problem and its opponent solves a bandit optimization problem, which we here call Markov-Bandit games. We extend Q-learning to solve Markov-Bandit games and show that our new Q-learning algorithms converge to the optimal solutions of the zero-sum Markov-Bandit games, and hence converge to the optimal solutions of the constrained and multi-objective Markov decision problems. We provide numerical examples where we calculate the optimal policies and show by simulations that the algorithm converges to the calculated optimal policies. To the best of our knowledge, this is the first time Q-learning algorithms guarantee convergence to optimal stationary policies for the multi-objective Reinforcement Learning problem with discounted and expected average rewards, respectively.

1 Introduction

1.1 Motivation

Reinforcement learning has made great advances in several applications, ranging from online learning and recommender engines, natural language understanding and generation, to mastering games such as Go (Silver et al., 2017) and Chess. The idea is to learn from extensive experience how to take actions that maximize a given reward by interacting with the surrounding environment. The interaction teaches the agent how to maximize its reward without knowing the underlying dynamics of the process. A classical example is swinging up a pendulum in an upright position. By making several attempts to swing up a pendulum and balancing it, one might be able to learn the necessary forces that need to be applied in order to balance the pendulum without knowing the physical model behind it, which is the general approach of classical model based control theory (Åström and Wittenmark, 1994).

Informally, the problem of constrained reinforcement learning for Markov decision processes is described as follows. Given a stochastic process with state s_k at time step k, reward function r, constraint function r^j, and a discount factor $0 < \gamma < 1$, the multi-objective reinforcement learning problem is that for the optimizing agent to find a stationary policy $\pi(s_k)$ that simultaneously satisfies in the discounted reward setting

$$\max_{\pi} \quad \mathbf{E}\left(\sum_{k=0}^{\infty} \gamma^k r(s_k, \pi(s_k))\right) \tag{1}$$

$$s.t. \quad \mathbf{E}\left(\sum_{k=0}^{\infty} \gamma^k r^j(s_k, \pi(s_k))\right) \geq 0 \tag{2}$$

or in the expected average reward setting

$$\max_{\pi} \quad \lim_{T\to\infty} \mathbf{E}\left(\frac{1}{T}\sum_{k=0}^{T-1} r(s_k, \pi(s_k))\right) \tag{3}$$

$$s.t. \quad \lim_{T\to\infty} \mathbf{E}\left(\frac{1}{T}\sum_{k=0}^{T-1} r^j(s_k, \pi(s_k))\right) \geq 0 \tag{4}$$

for $j = 1, ..., J$ (a more formal definition of the problem is introduced in the next section and some examples of this setup are given in Appendix.)

Surprisingly, although constrained MDP problems are fundamental and have been studied extensively in the literature (see (Altman, 1999) and the references therein), the reinforcement learning counter part seem to be still open. When an agent takes actions based on the observed states and constraint-outputs solely (without any knowledge about the dynamics, and/or

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

constraint-functions), a general solution seem to be lacking to the best of the author's knowledge for both the discounted and expected average rewards cases.

Note that maximizing Eq. (1) is equivalent to maximizing δ subject to the constraint

$$\mathbf{E}\left(\sum_{k=0}^{\infty}\gamma^k r(s_k,\pi(s_k))\right)\geq\delta$$

Thus, one could always replace r with $r-(1-\gamma)\delta$ and obtain a constraint of the form (1). Similarly for the average reward case, one may replace r with $r-\delta$ to obtain a constraint of the form (3). Hence, we can run the bisection method with respect to δ and the problem in discounted setting will be transformed to find a policy π such that

$$\mathbf{E}\left(\sum_{k=0}^{\infty}\gamma^k r^j(s_k,\pi(s_k))\right)\geq 0 \tag{5}$$

where $j=0,1,...,J$ and $r^0=r-(1-\gamma)\delta$. Or in the average setting, find policy π such that

$$\lim_{T\to\infty}\mathbf{E}\left(\frac{1}{T}\sum_{k=0}^{T-1} r^j(s_k,\pi(s_k))\right)\geq 0 \tag{6}$$

where $r^0=r-\delta$. In this paper, we call problems (5) and (6) as multi-objective MDPs problems and propose the algorithm with Markov bandit game and prove the convergence of them.

1.2 Related Work

Constrained MDP problems are convex and hence one can convert the constrained MDP problem to an unconstrained zero-sum game where the objective is the Lagrangian of the optimization problem (Altman, 1999). However, when the dynamics and rewards are not known, it doesn't become apparent how to do it as the Lagrangian will itself become unkown to the optimizing agent. Previous work regarding constrained MDPs, when the dynamics of the stochastic process are not known, considers scalarization through weighted sums of the rewards, see (Roijers et al., 2013) and the references therein. Another approach is to consider Pareto optimality when multiple objectives are present (Zhou et al., 2020) and Yang et al. (2019). Notice that there may be multiple Pareto optimal points, and all these points will not satisfy all the constraints in general. Further, using any solution of min-max may not in general be Pareto optimal. Thus, the problem formulation is very different from the above papers which aims to achieve the Pareto front.

In (Geibel, 2006), the author considers a single constraint and allowing for randomized policies. However, no proofs of convergence are provided for the proposed sub-optimal algorithms. Sub-optimal solutions with convergence guarantees were provided in (Chow et al. 2017) for the single constraint problem, allowing for randomized polices. In (Borkar, 2005), an actor-critic sub-optimal algorithm is provided for one single constraint and it's claimed that it can generalize to an arbitrary number of constraints. Reinforcement learning based model-free solutions have been proposed for the problems without guarantees (Djonin and Krishnamurthy, 2007; Lizotte et al., 2010; Drugan and Nowe, 2013; Achiam et al., 2017; Abels et al., 2019; Raghu et al., 2019).

Recently, (Tessler et al., 2018) proposed a policy gradient algorithm with Lagrange multiplier in multi-time scale for discounted constrained reinforcement learning algorithm and proved that the policy converges to a feasible policy. (Efroni et al., 2020) found a feasible policy by using Lagrange multiplier and zero-sum game for reinforcement learning algorithm with convex constraints and discounted reward. Yu et al. (2019) (Paternain et al. 2019) showed that constrained reinforcement learning has zero duality gap, which provides a theoretical guarantee to policy gradient algorithms in the dual domain. In constrast, our paper does not use policy gradient based algorithms. (Zheng and Ratliff, 2020) proposed the C-UCRL algorithm which achieves sub-linear $O(T^{\frac{3}{4}}\sqrt{\log(T)/\delta})$ with probability $1-\delta$, while satisfiying the constraints. However, this algorithm needs the knowledge of the model dynamics. Brantley et al. (2020) proposed a model-based algorithm for tabular episodic reinforcement learning with concave rewards and convex constraints. Singh et al. (2020) modified the famous UCRL2 algorithm and proposed the model-based UCRL-CMDP algorithm to solve the CMDP problem and gave the sub-linear result. Efroni et al. (2020) proposed 4 algorithms for the constrained reinforcement learning problem in primal, dual or primal-dual domain and showed a sub-linear bound for regret and constraints violations. However, all these algorithms are model based. While our algorithm is model-free and scalable to continuous spaces. Ding et al. (2020b) employed the natural policy gradient method to solve the discounted infinite-horizon CMDP problem. It achieves the $\mathcal{O}(\frac{1}{\epsilon^2})$ convergence rate with respite to the concept of ϵ-optimal or ϵ-constraint violation. Despite that the algorithm is model-free, it still needs simulator to get samples. Ding et al. (2020a) proposed a model-free primal-dual algorithm without the simulator to solve the CMDP and gives the $\mathcal{O}(\sqrt{T})$ bound for both reward and constraint, which should be the state-of-the-art result in this problem. Shah and Borkar (2018) proposed a three time scale Q-learning based algorithm to solve a constraint satisfaction problem in the

expected average setting. In contrast, this paper provides the first single time-scale Q-learning algorithm for both discounted and expected average rewards.

1.3 Contributions

We consider the problem of optimization and learning for constrained Markov decision processes, for both discounted rewards and expected average rewards. We formulate the problems as zero-sum games where one player (the agent) solves a Markov decision problem and its opponent solves a bandit optimization problem, which we here call Markov-Bandit games which are interesting on their own. The opponent acts on a *finite* set (and not on a continuous space). This transformation is essential in order to achieve a tractable optimal algorithm. The reason is that using Lagrange duality without model knowledge requires infinite dimensional optimization in the learning algorithm since the Lagrange multipliers are continuous (compare to the intractability of a partially observable MDP, where the beliefs are continuous variables). We extend Q-learning to solve Markov-Bandit games and show that our new Q-learning algorithms converge to the optimal solutions of the zero-sum Markov-Bandit games, and hence converge to the optimal solutions of the multi-objective Markov decision problems. The proof techniques are different for solving the discounted and average rewards problems, respectively, where the latter becomes much more technically involved. We provide numerical examples where we calculate the optimal policies and show by simulations that the algorithm converges to the calculated optimal policies. To the best of our knowledge, this is the first time Q-learning algorithms guarantee convergence to optimal stationary policies for the constrained and multi-objective MDP problem with discounted and expected average rewards, respectively.

1.4 Outline

In the problem formulation (Section 2), we present a precise mathematical definition of the multi-objective problem for MDPs. Then, we give a brief introduction and some useful results to reinforcement learning with applications to zero-sum Markov-Bandit games (Section 3). Section 4 firstly shows the connection between the Markov Bandit Game and the multi-objective problem. Then, it presents the solution to the multi-objective reinforcement learning problem . We demonstrate the proposed algorithm by an example in Section 5 and we finally conclude the paper in Section 6. Most proofs and some numerical results are relegated to the Appendix.

2 Problem Formulation and Assumptions

Consider a Markov Decision Process (MDP) defined by the tuple (S, A, P), where $S = \{S_1, S_2, ..., S_n\}$ is a finite set of states, $A = \{A_1, A_2, ..., A_m\}$ is a finite set of actions taken by the agent, and $P : S \times A \times S \to [0, 1]$ is a transition function mapping each triple (s, a, s_+) to a probability given by

$$P(s, a, s_+) = \mathbf{Pr}(s_+ \mid s, a)$$

and hence,

$$\sum_{s_+ \in S} P(s, a, s_+) = 1, \quad \forall (s, a) \in S \times A.$$

where $s+$ is the next state for state s. Let Π be the set of policies that map a state $s \in S$ to a probability distribution of the actions with a probability assigned to each action $a \in A$, that is $\pi(s) = a$ with probability $\mathbf{Pr}(a|s)$. The agent's objective in the multi-objective reinforcement learning is concerned with finding a policy that satisfies a set of constraints of the form ((5)) or (6), for $s_0 = s \in S$, where $r^j : S \times A \to \mathbb{R}$ are bounded functions, for $j = 0, 1, ..., J$, possibly unknown to the agent. The parameter $\gamma \in (0, 1)$ is a discount factor which models how much weight to put on future rewards. The expectation is taken with respect to the randomness introduced by the policy π and the transition mapping P.

Definition 1 (Unichain MDP)**.** *An MDP is called unichain, if for each policy π the Markov chain induced by π is ergodic, i.e. each state is reachable from any other state.*

Unichain MDPs are usually considered in reinforcement learning problems with discounted rewards, since they guarantee that we learn the process dynamics from the initial states. Thus, for the discounted reward case we will make the following assumption.

Assumption 1 (Unichain MDP)**.** *The MDP (S, A, P) is assumed to be unichain.*

For the case of expected average reward, we will make a simpler assumption regarding the existence of a recurring state, a standard assumption in Markov decision process problems with expected average rewards to ensure that the expected reward is independent of the initial state.

Assumption 2. *There exists a state $s^* \in S$ which is recurrent for every stationary policy π played by the agent.*

Assumption 2 implies that $\mathbf{E}(r^j(s_k, a_k))$ is independent of the initial state at stationarity. Hence, the

constraint (3) is at stationarity equivalent to the inequality $\mathbf{E}(r^j(s_k, a_k)) \geq 0$, for all k. We will use this constraint in the sequel which turns out to be very useful in the game-theoretic approach to solve the problem.

Assumption 3. *The absolute values of the functions* $\{r^j\}_{j=0}^J$ *are bounded by some constant* c *known to the agent.*

The bounded reward function is a typical assumption in RL, Jin et al. (2018) and Ni et al. (2019), where the bound is known aprior.

3 Reinforcement Learning for Zero-Sum Markov-Bandit Games

A zero-sum Markov-Bandit game is defined by the tuple (S, A, O, P, R), where S, A and P are defined as in section 2, $O = \{o_1, o_2, ..., o_q\}$ is a finite set of actions made by the agent's *opponent*. Let Π be the set of policies $\pi(s)$ that map a state $s \in S$ to a probability distribution of the actions with a probability assigned to each action $a \in A$, that is $\pi(s) = a$ with probability $\mathbf{Pr}(a \mid s)$.

For the zero-sum Markov-Bandit game, we define the reward $R : S \times A \times O \to \mathbb{R}$ which is assumed to be bounded. The agent's objective is to maximize the minimum (average or discounted) reward obtained due to the opponent's malicious action. The difference between a zero-sum Markov game and a Markov-Bandit game is that the opponent's action doesn't affect the state and it chooses a constant action $o_k = o \in O$ for all time steps k. This will be made more precise in the following sections.

3.1 Discounted Rewards

Consider a zero-sum Markov-Bandit game where the agent is maximizing the total discounted reward given by

$$V(s) = \min_o \mathbf{E}\left(\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k, o)\right) \tag{7}$$

for the initial state $s_0 \in S$. Let $Q(s, a, o)$ be the expected reward of the agent taking action $a_0 = a \in A$ from state $s_0 = s$, and continuing with a policy π thereafter when the opponent takes a fixed action o. Note that this is different from zero-sum Markov games with discounted rewards (Littman, 1994), where the opponent's actions may vary over time, that is o_k is not a constant. Then for any stationary policy π, we have that

$$\begin{aligned} Q(s, a, o) &= R(s, a, o) + \mathbf{E}\left(\sum_{k=1}^{\infty} \gamma^k R(s_+, \pi(s_+), o)\right) \\ &= R(s, a, o) + \gamma \cdot \mathbf{E}\left(Q(s_+, \pi(s_+), o)\right) \end{aligned} \tag{8}$$

Equation (8) is known as the Bellman equation. The solution to (8), with respect to Q and the initial state s_0 that corresponds to the optimal policy $\pi^\star$, is denoted $Q^\star$. If we have the function $Q^\star$, then we can obtain the optimal policy $\pi^\star$ according to the equations

$$\begin{aligned} Q^\star(s, a, o) &= R(s, a, o) + \gamma \cdot \mathbf{E}\left(Q^\star(s_+, \pi^\star(s_+), o)\right) \\ (\pi^\star(s_0), o^\star) &= \arg\max_{\pi \in \Pi} \min_o \mathbf{E}\left(Q^\star(s_0, \pi(s_0), o)\right) \\ \pi^\star(s) &= \arg\max_{\pi \in \Pi} \mathbf{E}\left(Q^\star(s, \pi(s), o^\star)\right) \end{aligned} \tag{9}$$

which maximizes the total discounted reward

$$\min_o \mathbf{E}\left(\sum_{k=0}^{\infty} \gamma^k R(s_k, \pi^\star(s), o)\right) = \min_o \mathbf{E}\left(Q^\star(s, \pi^\star(s), o)\right)$$

for $s = s_0$. Note that the optimal policy may not be deterministic, as opposed to reinforcement learning for unconstrained Markov Decision Processes, where there is always an optimal policy that is deterministic. Also, not that we will get different Q tables for different initial states here. Therefore, $Q^\star$ is in fact dependent on and varies with respect to s_0. A more proper notation would be to use $Q^\star_{s_0}$, but we omit the indexing with respect to s_0 for ease of notation. It's relevant to introduce the operator

$$(\mathbf{T}Q)(s, a, o) = R(s, a, o) + \gamma \cdot \mathbf{E}\left(Q(s_+, \pi^\star(s_+), o)\right)$$

which π^* appears in Equation (9). It's not hard to check that the operator $\mathbf{T}$ is not a contraction, so the standard Q-learning that is commonly used for reinforcement learning in Markov decision processes with discounted rewards can't be applied here.

In the case we don't know the process P and the reward function R, we will not be able to take advantage of the Bellman equation directly. The following results show that we will be able to design an algorithm that always converges to $Q^\star$.

Theorem 1. *Consider a zero-sum Markov-Bandit game given by the tuple* (S, A, O, P, R) *where* (S, A, P) *is unichain, and suppose that* R *is bounded by some constant and known aprior. Let* $Q = Q^\star$ *and* $\pi^\star$ *be solutions to*

$$\begin{aligned} Q(s, a, o) &= R(s, a, o) + \gamma \cdot \mathbf{E}\left(Q(s_+, \pi^\star(s_+), o)\right) \\ (\pi^\star(s), o^\star) &= \arg\max_{\pi \in \Pi} \min_o \mathbf{E}\left(Q(s, \pi(s), o)\right) \\ \pi^\star(s) &= \arg\max_{\pi \in \Pi} \mathbf{E}\left(Q(s, \pi(s), o^\star)\right) \end{aligned} \tag{10}$$

Let $\alpha_k(s,a,o) = \alpha_k \cdot 1_{(s,a,o)}(s_k,a_k,o_k)$ *satisfy*

$$0 \leq \alpha_k(s,a,o) < 1, \ \sum_{k=0}^{\infty} \alpha_k(s,a,o) = \infty, \\ \sum_{k=0}^{\infty} \alpha_k^2(s,a,o) < \infty, \ \forall (s,a,o) \in S \times A \times O. \tag{11}$$

Then, the update rule

$$(\pi_k, o_k) = \arg\max_{\pi \in \Pi} \min_{o} \mathbf{E}(Q_k(s_+, \pi(s_+), o))$$

$$Q_{k+1}(s,a,o_k) = \\ (1-\alpha_k(s,a,o_k))Q_k(s,a,o_k) + \alpha_k(s,a,o_k) \\ \times (R(s,a,o_k) + \gamma \mathbf{E}(Q_k(s_+, \pi_k(s_+), o_k))) \tag{12}$$

converges to $Q^\star$ *with probability 1. Furthermore, the optimal policy* $\pi^\star \in \Pi$ *given by (9) maximizes (7) with respect to the initial state* $s = s_0$*. That is,*

$$(\pi^\star(s_0), o^\star) = \arg\max_{\pi \in \Pi} \min_{o} \mathbf{E}\left(Q^\star(s_0, \pi(s_0), o)\right)$$

$$\pi^\star(s) = \arg\max_{\pi \in \Pi} \mathbf{E}\left(Q^\star(s, \pi(s), o^\star)\right)$$

3.2 Expected Average Rewards

The agent's objective is to maximize the minimal average reward obtained due to the opponent's malicious actions, that is maximizing the total reward given by

$$\min_{o \in O} \lim_{T \to \infty} \mathbf{E}\left(\frac{1}{T} \sum_{k=0}^{T-1} R(s_k, a_k, o)\right) \tag{13}$$

for some initial state $s_0 \in S$. Note that this problem is different from the zero-sum game considered in (Mannor, 2004a) where the opponent has to pick a fixed value for its action, $o_k = o$, as opposed to the work in (Mannor, 2004a) where o_k is allowed to vary over time. Thus, from the opponent's point of view, the opponent is performing bandit optimization.

Under Assumption 2 and for a given stationary policy π, the value of

$$V(o) \triangleq \lim_{T \to \infty} \mathbf{E}\left(\frac{1}{T} \sum_{k=0}^{T-1} R(s_k, \pi(s_k), o)\right) \tag{14}$$

is independent of the initial state s_0 for any fixed value of the parameter o. We will make this standard assumption in Markov decision process control problems.

Proposition 1. *Consider an MDP* (S, A, P) *with a total reward (14) for a fixed number* o*. Under Assumption 2 and for a fixed stationary policy* π*, there exists a number* $v(o)$ *and a vector* $H(s,o) = (H(S_1,o), ..., H(S_n,o)) \in \mathbb{R}^n$*, such that for each* $s \in S$*, we have that*

$$H(s,o) + v(o) = \mathbf{E}\Big(R(s,\pi(s),o) \\ + \sum_{s_+ \in S} P(s_+ \mid s, \pi(s)) H(s_+, o)\Big). \tag{15}$$

Furthermore, the value of (14) is $V(o) = v(o)$*.*

Proof. Consult (Bertsekas, 2005). □

Introduce

$$Q(s,a,o) - v(o) = R(s,a,o) + \sum_{s_+ \in S} P(s_+ \mid s, a) H(s_+, o) \tag{16}$$

and let $Q^\star$, $v^\star$, and $H^\star$ be solutions to Equation (15)-(16) corresponding to the optimal policy $\pi^\star$ that maximizes (13). Then we have that

$$\pi^\star(s) = \arg\max_{\pi \in \Pi} \min_{o \in O} \mathbf{E}\left(Q^\star(s, \pi(s), o)\right)$$

$$H^\star(s,o) = \mathbf{E}\left(Q^\star(s, \pi^\star(s), o)\right)$$

$$Q^\star(s,a,o) - v^\star(o) \\ = R(s,a,o) + \sum_{s_+ \in S} P(s_+ \mid s, \pi^\star(s)) H^\star(s_+, o) \tag{17}$$

We will make some additional assumptions that will be used in the learning of $Q^\star$ in the average reward case. We start off by introducing a sequence of learning rates $\{\beta_k\}$ and assume that this sequence satisfies the following assumption. Notice that this is a typical assumption in stochastic approximation and it is common in expected reward RL setting. See for instance Assumption 2.3 in Abounadi et al. (2001a) and Assumption 2 in Mannor (2004b).

Assumption 4 (Learning rate)**.** *The sequence* β_k *satisfies:*

1. $\beta_{k+1} \leq \beta_k$ *eventually*
2. *For every* $0 < x < 1$, $\sup_k \beta_{\lfloor xk \rfloor}/\beta_k < \infty$
3. $\sum_{k=1}^{\infty} \beta_k = \infty$ *and* $\sum_{k=0}^{\infty} \beta_k^2 < \infty$.
4. *For every* $0 < x < 1$*, the fraction*

$$\frac{\sum_{k=1}^{\lfloor yt \rfloor} \beta_k}{\sum_{k=1}^{t} \beta_k}$$

converges to 1 uniformly in $y \in [x, 1]$ *as* $t \to \infty$*.*

For example, $\beta_k = \frac{1}{k}$ and $\beta_k = \frac{1}{k \log k}$ (for $k > 1$) satisfy Assumption 4.

Now define $N(k,s,a,o)$ as the number of times that state s and actions a and o were played up to time k, that is

$$N(k,s,a,o) = \sum_{t=1}^{k} 1_{(s,a,o)}(s_t,a_t,o_t).$$

The following assumption is needed to guarantee that all combinations of the triple (s,a,o) are visited often, which can be satisfied by using the ϵ-greedy method in the algorithm.

Assumption 5 (Often updates)**.** *There exists a deterministic number $d > 0$ such that for every $s \in S$, $a \in A$, and $o \in O$, we have that*

$$\liminf_{k\to\infty} \frac{N(k,s,a,o)}{k} \geq d$$

with probability 1.

Definition 2. *We define the set Φ as the set of all functions $f : \mathbb{R}^{n\times m\times q} \to \mathbb{R}$ such that*

1. *f is Lipschitz*
2. *For any $c \in \mathbb{R}$, $f(cQ) = cf(Q)$*
3. *For any $r \in \mathbb{R}$ and $\widehat{Q}(s,a,o) = Q(s,a,o) + r$ for all $(s,a,o) \in \mathbb{R}^{n\times m\times q}$, we have $f(\widehat{Q}) = f(Q) + r$*

For instance, $f(Q) = \frac{1}{|S||A||O|}\sum_{s,a,o} Q(s,a,o)$ belongs the the set Φ.

The next result shows that we will be able to design an algorithm that always converges to $Q^\star$.

Theorem 2. *Consider a Markov-Bandit zero-sum game given by the tuple (S,A,O,P,R) and suppose that R is bounded. Suppose that Assumption 2, 4, and 5 hold. Let $f \in \Phi$ be given, where the set Φ is defined as in Definition 2. Then, the asynchronous update algorithm given by*

$$\begin{aligned} Q_{k+1}(s,a,o) &= Q_k(s,a,o) + 1_{(s,a,o)}(s_k,a_k,o_k)\times \\ &\times \beta_{N(k,s,a,o)} \max_{\pi\in\Pi} \min_{o_k\in O}(R(s,a,o_k) \\ &+ \mathbf{E}(Q_k(s_{k+1},\pi(s_{k+1}),o_k)) - f(Q_k) - Q_k(s,a,o)) \end{aligned} \tag{18}$$

converges to $Q^\star$ in (17) with probability 1. Furthermore, the optimal policy $\pi^\star \in \Pi$ given by (17) maximizes (13).

4 Reinforcement Learning for multi-objective Reinforcement Learning

4.1 Discounted Rewards

Consider the optimization problem of finding a stationary policy π subject to the initial state $s_0 = s$ and the constraints (1), that is

$$\begin{aligned} &\text{find} && \pi \in \Pi \\ &\text{s. t.} && \mathbf{E}\left(\sum_{k=0}^{\infty} \gamma^k r^j(s_k,\pi(s_k))\right) \geq 0 \\ &&& \text{for } j = 1,...,J. \end{aligned} \tag{19}$$

The next theorem states that the optimization problem (19) is equivalent to a zero-sum Markov-Bandit game, in the sense that an optimal strategy of the agent in the zero-sum game is also optimal for (19).

Theorem 3. *Consider optimization problem (19) and suppose it's feasible and that Assumption 3 holds. Let $\pi^\star$ be an optimal stationary policy in the zero-sum game*

$$v(s_0) = \max_{\pi\in\Pi} \min_{j\in[J]} \mathbf{E}\left(\sum_{k=0}^{\infty} \gamma^k r^j(s_k,\pi(s_k))\right). \tag{20}$$

Then, $\pi^\star$ is a feasible solution to (19) if and only if $v(s_0) \geq 0$.

The interpretation of the game (20) is that the minimizer chooses index $j \in [J]$, where $[J] = 1,2,...,J$.

Now that we are equipped with Theorem 1 and 3, we are ready to state and prove our next result.

Theorem 4. *Consider the constrained MDP problem (19) and suppose that it's feasible and that Assumption 1 and 3 hold. Also, introduce $O = [J]$, $o = j$, and*

$$R(s,a,o) = R(s,a,j) \triangleq r^j(s,a), \qquad j = 1,...,J.$$

Let Q_k be given by the recursion according to (12). Then, $Q_k \to Q^\star$ as $k \to \infty$ where $Q^\star$ is the solution to (10). Furthermore, the policy

$$\begin{aligned} \pi^\star(s_0) &= \arg\max_{\pi\in\Pi} \min_{o\in O} \mathbf{E}\left(Q^\star(s_0,\pi(s),o)\right) \\ \pi^\star(s) &= \arg\max_{\pi\in\Pi} \mathbf{E}(Q^\star(s,\pi(s),o^\star)) \end{aligned} \tag{21}$$

is an optimal solution to (19) for all $s \in S$.

Proof. According to Theorem 3, (19) is equivalent to the zero-sum game (20), which is equivalent to the zero-sum Markov-Bandit game given by (S,A,O,P,R) with the objective

$$\max_{\pi\in\Pi} \min_{o\in O} \mathbf{E}\left(\sum_{k=0}^{\infty} \gamma^k R(s_k,\pi(s_k),o)\right). \tag{22}$$

Assumption 3 implies that $|R(s,a,o)| \leq 2c$ for all $(s,a,o) \in S\times A\times O$. Now let $Q^\star$ be the solution to the maximin Bellman equation (10). According to Theorem 1, Q_k in the recursion given by (11)-(12) converges

Algorithm 1 Zero Sum Markov Bandit Algorithm for CMDP with Discounted Reward

1: Initialize $Q(s,a,o) \leftarrow 0$ for all $(s,a,o) \in \mathcal{S} \times \mathcal{A} \times \mathcal{O}$. Observe s_0 and Initial a_0 randomly. Select α_k according to Eq. (11)
2: **for** Iteration $k = 0, ..., K$ **do**
3: Take action a_k and observe next state s_{k+1}
4: $\pi_{k+1}, o_k = \arg\max_{\pi_{k+1}} \min_{o \in \mathcal{O}} Q(s_{k+1}, \pi_{k+1}(s_{k+1}), o_k)$
5: $Q(s_k,a_k,o_k) \leftarrow (1-\alpha_k)Q(s_k,a_k,o_k) + \alpha_k[r(s_k,a_k,o_k) + \gamma \mathbf{E}(Q(s_{k+1}, \pi_{k+1}(s_{k+1}), o_k)]$
6: Sample a_{k+1} from the distribution $\pi_{k+1}(\cdot|s_{k+1})$
7: **end for**

to $Q^\star$ with probability 1. By, definition, the optimal policy $\pi^\star$ achieves the value of the zero-sum Markov-Bandit game in (21), and thus achieves the value of (22) and the proof is complete. □

Finally, the algorithm for Constrained Markov Decision Process with Discounted Reward is shown in Alg. 1. In line 1, we initialize the Q-table, observe s_0 and select a_0 randomly. In line 3, we take the current action a_k and observe the next state s_{k+1} so that we can compute the max-min operator in line 4 based on the first line of Eq. (12). Line 5 updates the Q-table according to the second line of Eq. (12). Line 6 samples the next action from the policy gotten from the line 4. Notice that The max-min can be converted to a maximization problem with linear inequalities and can be solved by linear programming efficiently due to the number of inequalities here is limited by the space of the opponent. Even for large scale problems, efficient algorithms exist for max-min Parpas and Rustem (2009).

4.2 Expected Average Rewards

Consider the optimization problem of finding a stationary policy π subject to the constraints (3), that is

$$\begin{aligned} &\text{find} \quad \pi \in \Pi \\ &\text{s. t.} \quad \lim_{T\to\infty} \mathbf{E}\left(\frac{1}{T}\sum_{k=0}^{T-1} r^j(s_k, \pi(s_k))\right) \geq 0 \qquad (23) \\ &\qquad \text{for } j = 1, ..., J. \end{aligned}$$

The next theorem states that the optimization problem (23) is equivalent to a zero-sum Markov-Bandit game, in the sense that an optimal strategy of the agent in the zero-sum game is also optimal for (23).

Theorem 5. *Consider optimization problem (23) and suppose that Assumption 2 and 3 hold. Let $\pi^\star$ be an optimal policy in the zero-sum game*

$$v = \max_{\pi\in\Pi} \min_{j\in[J]} \lim_{T\to\infty} \mathbf{E}\left(\frac{1}{T}\sum_{k=0}^{T-1} r^j(s_k, \pi(s_k))\right). \quad (24)$$

Then, $\pi^\star$ is a solution to (23) if and only if $v \geq 0$.

Algorithm 2 Zero Sum Markov Bandit Algorithm for CMDP with Average Reward

1: Initialize $Q(s,a,o) \leftarrow 0$ and $N(s,a,o) \leftarrow 0 \ \forall (s,a,o) \in \mathcal{S} \times \mathcal{A} \times \mathcal{O}$. Observe s_0 and initialize a_0 randomly
2: **for** Iteration $k = 0, ..., K$ **do**
3: Take action a_k and observe next state s_{k+1}
4: $\pi_{k+1}, o_k = \arg\max_{\pi_{k+1}} \min_{o\in\mathcal{O}} \Big[R(s_k,a_k,o_k) + Q(s_{k+1}, \pi_{k+1}(s_{k+1}), o_k)\Big]$
5: $t = N(s_k,a_k,o_k) \leftarrow N(s_k,a_k,o_k) + 1$; $\alpha_t = \frac{1}{t+1}$
6: $f = \frac{1}{|S||A|O|}\sum_{s,a,o} Q(s,a,o)$
7: $y = R(s_k,a_k,o_k) + \mathbf{E}[Q(s_{k+1}, \pi_{k+1}(s_{k+1}), o_k] - f$
8: $Q(s_k,a_k,o_k) \leftarrow (1-\alpha_t)Q(s_k,a_k,o_k) + \alpha_k * y$
9: Sample a_{k+1} from the distribution $\pi_{k+1}(\cdot|s_{k+1})$
10: **end for**

Now that we are equipped with Theorem 2 and 5, we are ready to state the second main result (proof in Appendix).

Theorem 6. *Consider the constrained Markov Decision Process problem (23) and suppose that Assumption 2 and 3 hold. Introduce $O = [J]$, $o = j$ and*

$$R(s,a,o) = R(s,a,j) \triangleq r^j(s,a), \qquad j = 1, ..., J$$

Let Q_k be given by the recursion according to (18) and suppose that Assumptions 4 and 5 hold. Then, $Q_k \to Q^\star$ as $k \to \infty$ where $Q^\star$ is the solution to (17). Furthermore, the policy

$$\pi^\star(s) = \arg\max_{\pi\in\Pi} \min_{o\in O} \mathbf{E}\left(Q^\star(s, \pi(s), o)\right) \qquad (25)$$

is a solution to (23) for all $s \in S$.

The algorithm for Constrained Markov Decision Process with Discounted Reward is in Alg. 2. The most part of this algorithm is similar to Algorithm 1. However, in line 1, we initialize the N table, which records how many times (s,a,o) has been met in the learning process and N table is updated in line 5. Besides, in Line 6, f is computed according to Def. 2. Finally, in line 8, Q-table is updated according to the Eq. (2).

Table 1: Transition probability of the queue system

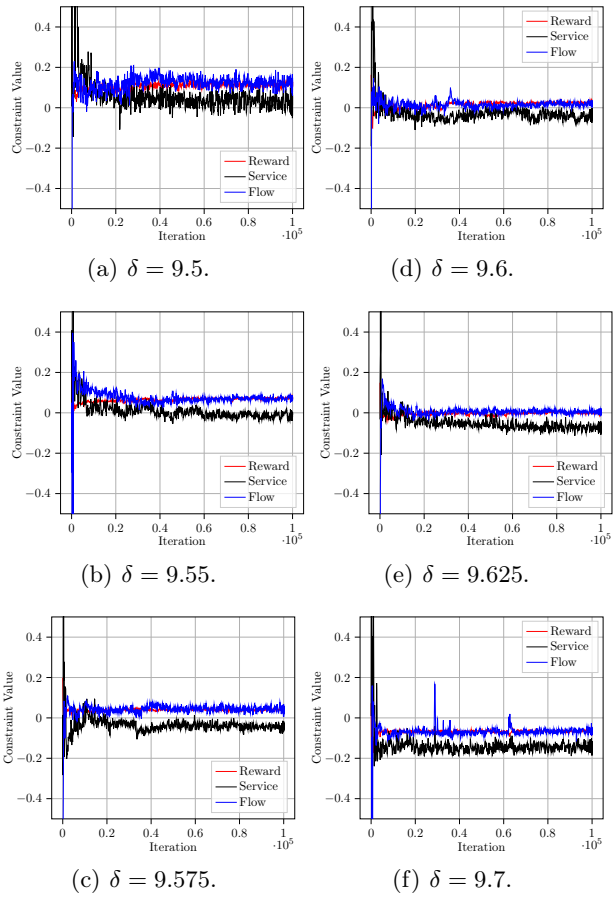
Current State	$P(x_{t+1} = x_t - 1)$	$P(x_{t+1} = x_t)$	$P(x_{t+1} = x_t + 1)$
$1 \le x_t \le L-1$	$a(1-b)$	$ab + (1-a)(1-b)$	$(1-a)b$
$x_t = L$	a	$1-a$	0
$x_t = 0$	0	$1-b(1-a)$	$b(1-a)$

5 Evaluation on Discrete time Single-Server Queue

In this section, we evaluate the proposed algorithm on a queuing system with a single server in discrete time. In this model, we assume there is a buffer of finite size L. A possible arrival is assumed to occur at the beginning of the time slot. The state of the system is the number of customers waiting in the queue at the beginning of time slot such that $|S| = L + 1$. We assume there are two kinds of actions, service action and flow action. The service action space is a finite subset A of $[a_{min}, a_{max}]$ and $0 < a_{min} \le a_{max} < 1$. With a service action a, we assume that a service of a customer is successfully completed with probability a. If the service succeeds, the length of the queue will reduce by one, otherwise there is no change of the queue. The flow is a finite subset B of $[b_{min}, b_{max}]$ and $0 \le b_{min} \le b_{max} < 1$. Given a flow action b, a customer arrives during the time slot with probability b. Let the state at time t be x_t. We assume that no customer arrives when state $x_t = L$ and thus can model this by the state update not increasing on customer arrival when $x_t = L$. Finally, the overall action space is the product of service action space and flow action space, i.e., $A \times B$. Given an action pair (a, b) and current state x_t, the transition of this system $P(x_{t+1}|x_t, a_t = a, b_t = b)$ is shown in Table 1. Assuming that $\gamma = 0.5$, we want to optimize the total discounted reward collected and satisfies two constraints with respect to service and flow simultaneously. Thus, the overall optimization problem is given as

$$\begin{aligned} \min_{\pi^a, \pi^b} \quad & \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, \pi^a(s_t), \pi^b(s_t))\right] \\ s.t. \quad & \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c^i(s_t, \pi^a(s_t), \pi^b(s_t))\right] \le 0 \quad i = 1, 2 \end{aligned} \tag{26}$$

where π_h^a and π_h^b are the policies for the service and flow at time slot h, respectively. We note that the expectation in the above is with respect to both the stochastic policies and the transition probability. In order to match the constraints satisfaction problem modeled in this paper, we use the bisection algorithm on δ and transform the objective to a constraint $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, \pi^a(s_t), \pi^b(s_t))\right] \le \delta$. In the setting of the simulation, we choose the length of the queue $L = 5$. We let the service action space be $A = [0.3, 0.4, 0.5, 0.6, 0.7]$ and the flow action space be $B = [0, 0.2, 0.4, 0.6]$ for all states besides the state $s = L$. Moreover, the cost function is set to be $c(s, a, b) = s - 5$, the constraint function for the service is defined as $c^1(s, a, b) = 10a - 5$, and the constraint function for the flow is $c^2(s, a, b) = 5(1-b)^2 - 2$.



(a) $\delta = 9.5$. (d) $\delta = 9.6$.

(b) $\delta = 9.55$. (e) $\delta = 9.625$.

(c) $\delta = 9.575$. (f) $\delta = 9.7$.

Figure 1: Value of constraints with iterations for the Zero-Sum Markov Bandit algorithm applied to Discrete time single-server queue in the discounted case.

For different values of δ, the numerical results are given in Fig. 1. To show the performance of the algorithm, we choose the values of δ close to the real optimum value and thus the figure shows the performance with $\delta = 9.5$, 9.55, 9.575, 9.6, 9.625, and 9.7. For each value of δ, we run the algorithm for 10^5 iterations. Rather than evaluating the policy in each iteration, we evaluate the policy every 100 iterations, while evaluate at every iteration for the last 100 iterations. In order to get the expected value of the constraints, we collect 10000 trajectories and calculate the average constraint function value among them. These constraint function values for the three constraints are plotted in Fig. 1. For $\delta = 9.5$, we see that the algorithm converges after about 60000 iterations and all three constrains are larger than 0, which means that we find a feasible policy for the setting $\delta = 9.5$. Moreover, it is reasonable that all three constraints converge to a same value

since the proposed Algorithm 1 optimize the minimal value function among $V(s,a,o)$ with respect to o. We see that the three constraints for $\delta = 9.5$ are close to each other and non-negative, thus demonstrating the constraints are satisfied and $\delta = 9.5$ is feasible.

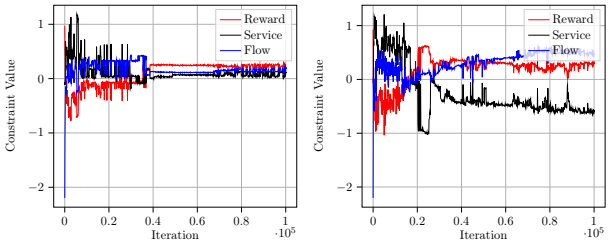
On the other extreme, we see the case when $\delta = 9.7$. We note that all three constraints are below 0, which means that no feasible policy for this setting. Thus, seeing the cases for $\delta = 9.5$ and 9.7, we note that the optimal objective is between the two values. Looking at the case where $\delta = 9.625$, we also note that the service constraint is clearly below zero and the constraints are not satisfied. Similarly, for $\delta = 9.55$, the constraints are non-negative - the closest to zero are the service constraints which are crossing zero every few iterations and thus the gap is within the margin. This shows that the optimal objective is within 9.55 and 9.625. However, the judgment is not as evident between the two regimes and it cannot be clearly mentioned from $\delta = 9.575$ and $\delta = 9.6$ if they are feasible or not since they are not consistently lower than zero after 80,000 iterations like in the case of $\delta = 9.625$ and $\delta = 9.7$, are not mostly above zero as for $\delta = 9.5$. Thus, looking at the figures, we estimate the value of optimal objective between 9.55 and 9.625.

In order to compare the result with the theoretical optimal total reward, we can assume the dynamics of the MDP is known in advance and use the Linear Programming algorithm to solve the original problem. The result solved by the LP is 9.62. We note that $Q(s,a,o)$ has $6\times5\times4\times3 = 360$ elements and it is possible that 10^5 iterations are not enough to make all the elements in Q table to converge. Further, sampling 10^4 trajectories can only achieve an accuracy of 0.1 with 99% confidence for the constraint function value and we are within that range. Thus, more iterations and more samples (especially more samples) would help improve the achievable estimate from 9.55 in the algorithm performance. Overall, considering the limited iterations and sampling in the simulations, we conclude that the result by the proposed algorithm is close to the optimal result obtained by the Linear Programming.

Next, we test this example in the expected average rewards case, which is formulated as

$$\begin{aligned} \min_{\pi^a,\pi^b} \quad & \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} c(s_t,\pi^a(s_t),\pi^b(s_t))\right] \\ s.t. \quad & \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} c^i(s_t,\pi^a(s_t),\pi^b(s_t))\right] \le 0 \quad i=1,2 \end{aligned} \tag{27}$$

When the model is known apriori, this problem can be computed by linear programming approach (Altman, 1999). With the same reward function and two constraint functions, LP shows that the optimal total average reward is about 3.62. In order to compare the result with LP, we run Algorithm 2 with parameters $\delta = 3.5$ and $\delta = 3.55$. The results are shown in Fig. 2.



(a) $\delta = 3.5$. (b) $\delta = 3.55$.

Figure 2: Value of constraints with iterations for the Zero-Sum Markov Bandit algorithm applied to Discrete time single-server queue in the expected average case.

We see that for $\delta = 3.5$, all the three constraints are all above 0, which means that there exists feasible policy when $\delta = 3.5$. However, when $\delta = 3.55$, the flow constraint is not satisfied. Thus, the result by the proposed algorithm is between between 3.5 and 3.55, which is close to the optimal result 3.62 (after converting objective to a constraint). The reason for the gap arises from the same reason as mentioned in the discounted case.

Finally, We note that two additional numerical examples can be seen in the Appendix.

6 Conclusions

We considered the problem of optimization and learning for constrained and multi-objective Markov decision processes, for both discounted rewards and expected average rewards. We formulated the problems as zero-sum games where one player (the agent) solves a Markov decision problem and its opponent solves a bandit optimization problem, which we call Markov-Bandit games. We extended Q-learning to solve Markov-Bandit games and proved that our new Q-learning algorithms converge to the optimal solutions of the zero-sum Markov-Bandit games, and hence converge to the optimal solutions of the constrained and multi-objective Markov decision problems. The provided numerical examples and the simulation results illustrate that the proposed algorithm converges to the optimal policy.

Having an approach with a combination of the long-term constraints as in this paper and the peak constraints as in Gattami (2019); Bai et al. (2021) is an interesting future direction.

References

Abels A, Roijers D, Lenaerts T, Nowé A, Steckelmacher D (2019) Dynamic weights in multi-objective deep reinforcement learning. In: Chaudhuri K, Salakhutdinov R (eds) Proceedings of the 36th International Conference on Machine Learning, PMLR, Long Beach, California, USA, Proceedings of Machine Learning Research, vol 97, pp 11–20, URL http://proceedings.mlr.press/v97/abels19a.html

Abounadi J, Bertsekas D, Borkar V (2001a) Learning algorithms for markov decision processes with average cost. SIAM J Control Optim 40:681–698

Abounadi J, Bertsekas D, Borkar VS (2001b) Learning algorithms for markov decision processes with average cost. SIAM J Control and Optimization 40:681–698

Achiam J, Held D, Tamar A, Abbeel P (2017) Constrained policy optimization. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70, JMLR.org, ICML'17, pp 22–31, URL http://dl.acm.org/citation.cfm?id=3305381.3305384

Altman E (1999) Constrained Markov decision processes, vol 7. CRC Press

Åström KJ, Wittenmark B (1994) Adaptive Control, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA

Bai Q, Aggarwal V, Gattami A (2021) Provably efficient model-free algorithm for mdps with peak constraints. arXiv preprint arXiv:200305555

Bertsekas DP (2005) Dynamic Programming and Optimal Control, vol 1. Athena Scientific

Borkar VS (2005) An actor-critic algorithm for constrained markov decision processes. Systems & Control Letters 54(3):207 – 213, DOI https://doi.org/10.1016/j.sysconle.2004.08.007, URL http://www.sciencedirect.com/science/article/pii/S0167691104001276

Brantley K, Dudik M, Lykouris T, Miryoosefi S, Simchowitz M, Slivkins A, Sun W (2020) Constrained episodic reinforcement learning in concave-convex and knapsack settings. arXiv preprint arXiv:200605051

Chow Y, Ghavamzadeh M, Janson L, Pavone M (2017) Risk-constrained reinforcement learning with percentile risk criteria. Journal of Machine Learning Research 18:167:1–167:51

Ding D, Wei X, Yang Z, Wang Z, Jovanović MR (2020a) Provably efficient safe exploration via primal-dual policy optimization. arXiv preprint arXiv:200300534

Ding D, Zhang K, Jovanovic M, Basar T (2020b) Natural policy gradient primal-dual method for constrained markov decision processes. NIPS 2020

Djonin DV, Krishnamurthy V (2007) Mimo transmission control in fading channels: A constrained markov decision process formulation with monotone randomized policies. IEEE Transactions on Signal Processing 55(10):5069–5083, DOI 10.1109/TSP.2007.897859

Drugan MM, Nowe A (2013) Designing multi-objective multi-armed bandits algorithms: A study. In: The 2013 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–8

Efroni Y, Mannor S, Pirotta M (2020) Exploration-exploitation in constrained mdps. arXiv preprint arXiv:200302189

Gattami A (2019) Reinforcement learning of markov decision processes with peak constraints. arXiv preprint arXiv:190107839

Geibel P (2006) Reinforcement learning for MDPs with constraints. In: Fürnkranz J, Scheffer T, Spiliopoulou M (eds) Machine Learning: ECML 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 646–653

Jaakkola T, Jordan MI, Singh SP (1994) On the convergence of stochastic iterative dynamic programming algorithms. Neural Comput 6(6):1185–1201, DOI 10.1162/neco.1994.6.6.1185, URL http://dx.doi.org/10.1162/neco.1994.6.6.1185

Jin C, Allen-Zhu Z, Bubeck S, Jordan MI (2018) Is q-learning provably efficient? In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, NIPS'18, p 4868–4878

Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the Eleventh International Conference on International Conference on Machine Learning, San Francisco, CA, USA, pp 157–163, URL http://dl.acm.org/citation.cfm?id=3091574.3091594

Lizotte D, Bowling MH, Murphy AS (2010) Efficient reinforcement learning with multiple reward functions for randomized controlled trial analysis. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, Omnipress, USA, ICML'10, pp 695–702, URL http://dl.acm.org/citation.cfm?id=3104322.3104411

Mannor S (2004a) Reinforcement learning for average reward zero-sum games. In: Shawe-Taylor J, Singer Y (eds) Learning Theory, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 49–63

Mannor S (2004b) Reinforcement learning for average reward zero-sum games. In: Shawe-Taylor J, Singer Y (eds) Learning Theory, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 49–63

Ni C, Yang LF, Wang M (2019) Learning to control in metric space with optimal regret. In: 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE Press, p 726–733, DOI 10.1109/ALLERTON.2019.8919864, URL https://doi.org/10.1109/ALLERTON.2019.8919864

Parpas P, Rustem B (2009) An algorithm for the global optimization of a class of continuous minimax problems. Journal of Optimization Theory and Applications 141:461–473, DOI 10.1007/s10957-008-9473-4

Paternain S, Chamon LF, Calvo-Fullana M, Ribeiro A (2019) Constrained reinforcement learning has zero duality gap. arXiv preprint arXiv:191013393

Raghu R, Upadhyaya P, Panju M, Agarwal V, Sharma V (2019) Deep reinforcement learning based power control for wireless multicast systems. In: 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, pp 1168–1175

Roijers DM, Vamplew P, Whiteson S, Dazeley R (2013) A survey of multi-objective sequential decision-making. J Artif Int Res 48(1):67–113, URL http://dl.acm.org/citation.cfm?id=2591248.2591251

Shah SM, Borkar VS (2018) Q-learning for markov decision processes with a satisfiability criterion. Systems and Control Letters 113:45–51, DOI https://doi.org/10.1016/j.sysconle.2018.01.003, URL https://www.sciencedirect.com/science/article/pii/S0167691118300045

Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y, Lillicrap T, Hui F, Sifre L, Driessche GVD, Graepel T, Hassabis D (2017) Mastering the game of go without human knowledge. Nature 550:354 – 359, URL http://dx.doi.org/10.1038/nature24270

Singh R, Gupta A, Shroff NB (2020) Learning in markov decision processes under constraints. arXiv preprint arXiv:200212435

Tessler C, Mankowitz DJ, Mannor S (2018) Reward constrained policy optimization. In: International Conference on Learning Representations

Yang R, Sun X, Narasimhan K (2019) A generalized algorithm for multi-objective reinforcement learning and policy adaptation. arXiv preprint arXiv:190808342

Yu M, Yang Z, Kolar M, Wang Z (2019) Convergent policy optimization for safe reinforcement learning. Advances in Neural Information Processing Systems 32

Zheng L, Ratliff L (2020) Constrained upper confidence reinforcement learning. PMLR, The Cloud, Proceedings of Machine Learning Research, vol 120, pp 620–629, URL http://proceedings.mlr.press/v120/zheng20a.html

Zhou D, Chen J, Gu Q (2020) Provable multi-objective reinforcement learning with generative models. arXiv preprint arXiv:201110134