Online Learning in Weakly Coupled Markov Decision Processes: A Convergence Time Study

Xiaohan Wei
University of Southern California
Los Angeles, CA, USA
xiaohanw@usc.edu

Hao Yu
University of Southern California
Los Angeles, CA, USA
yuhao@usc.edu

Michael J. Neely
University of Southern California
Los Angeles, CA, USA
mikejneely@gmail.com

ABSTRACT

We consider multiple parallel Markov decision processes (MDPs) coupled by global constraints, where the time varying objective and constraint functions can only be observed *after* the decision is made. Special attention is given to how well the decision maker can perform in T slots, starting from any state, compared to the best feasible randomized stationary policy in hindsight. We develop a new distributed online algorithm where each MDP makes its own decision each slot after observing a multiplier computed from past information. While the scenario is significantly more challenging than the classical online learning context, the algorithm is shown to have a tight $O(\sqrt{T})$ regret and constraint violations simultaneously. To obtain such a bound, we combine several new ingredients including ergodicity and mixing time bound in weakly coupled MDPs, a new regret analysis for online constrained optimization, a drift analysis for queue processes, and a perturbation analysis based on Farkas' Lemma.

CCS CONCEPTS

• **Theory of computation** → **Probabilistic computation**; *Streaming models*;

KEYWORDS

Stochastic programming; Online learning; Markov decision processes

ACM Reference Format:

Xiaohan Wei, Hao Yu, and Michael J. Neely. 2018. Online Learning in Weakly Coupled Markov Decision Processes: A Convergence Time Study. In *SIGMETRICS'18 Abstracts: ACM SIGMETRICS International Conference on Measurement & Modeling of Computer Systems Abstracts, June 18–22, 2018, Irvine, CA, USA*. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3179415>

1 INTRODUCTION

This paper considers online constrained Markov decision processes (OCMDP) where both the objective and constraint functions can vary each time slot after the decision is made. We assume a slot-*t* time scenario with time slots $t \in \{0, 1, 2, \dots\}$. The OCMDP consists of K parallel Markov decision processes with indices $k \in \{1, 2, \dots, K\}$. The k -th MDP has state space $\mathcal{S}^{(k)}$, action space $\mathcal{A}^{(k)}$,

and transition probability matrix $P_a^{(k)}$ which depends on the chosen action $a \in \mathcal{A}^{(k)}$. Specifically, $P_a^{(k)} = (P_a^{(k)}(s, s'))$ where

$$P_a^{(k)}(s, s') = \Pr(s_{t+1}^{(k)} = s' \mid s_t^{(k)} = s, a_t^{(k)} = a),$$

where $s_t^{(k)}$ and $a_t^{(k)}$ are the state and action for system k on slot t . We assume that both the state space and the action space are finite for all $k \in \{1, 2, \dots, K\}$.

After each MDP $k \in \{1, \dots, K\}$ makes the decision at time t (and assuming the current state is $s_t^{(k)} = s$ and the action is $a_t^{(k)} = a$), the following information is revealed:

- (1) The next state $s_{t+1}^{(k)}$.
- (2) A penalty function $f_t^{(k)}(s, a)$ that depends on the current state s and the current action a .
- (3) A collection of m constraint functions $g_{1,t}^{(k)}(s, a), \dots, g_{m,t}^{(k)}(s, a)$ that depend on s and a .

The functions $f_t^{(k)}$ and $g_{i,t}^{(k)}$ are all bounded mappings from $\mathcal{S}^{(k)} \times \mathcal{A}^{(k)}$ to $\mathbb{R}$ and represent different types of costs incurred by system k on slot t (depending on the current state and action). Let $\mathbf{f}^{(k)} := [f^{(k)}(a, s)]_{a \in \mathcal{A}, s \in \mathcal{S}}$ and $\mathbf{g}_i^{(k)} := [g_i^{(k)}(a, s)]_{a \in \mathcal{A}, s \in \mathcal{S}}$. Note that in our model, the functions $f_t^{(k)}$ are *arbitrary* time-varying processes with no assumed probability structure. The constraint functions $g_{i,t}^{(k)}$ are time-varying but are assumed to be i.i.d. over slots with unknown distributions.

One concrete example of the above model is a multi-server data center, where the different systems $k \in \{1, \dots, K\}$ can represent different servers, the penalty function for a particular server k can represent monetary expenditure for the power on that server, whose per unit price can change arbitrarily over time, and the constraints can represent service rate requirements on these servers to balance the job arrivals. Coupling among the server systems comes from using all of them to collectively support a common stream of arriving jobs.

A key aspect of this general problem is that the functions $f_t^{(k)}$ and $g_{i,t}^{(k)}$ are unknown until after the slot t decision is made. Thus, the precise costs incurred by each system are only known at the end of the slot. For any fixed time horizon of T slots, the overall penalty and constraint accumulation resulting from a policy $\mathcal{P}$ is:

$$F_T(d_0, \mathcal{P}) := \mathbb{E} \left(\sum_{t=1}^T \sum_{k=1}^K f_t^{(k)}(a_t^{(k)}, s_t^{(k)}) \mid d_0, \mathcal{P} \right), \quad (1)$$

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMETRICS'18 Abstracts, June 18–22, 2018, Irvine, CA, USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5846-0/18/06.

<https://doi.org/10.1145/3179415>

and

$$G_{i,T}(d_0, \mathcal{P}) := \mathbb{E} \left(\sum_{t=1}^T \sum_{k=1}^K g_{i,t}^{(k)} \left(a_t^{(k)}, s_t^{(k)} \right) \middle| d_0, \mathcal{P} \right),$$

where d_0 represents a given distribution on the initial joint state vector $(s_0^{(1)}, \dots, s_0^{(K)})$. Note that $(a_t^{(k)}, s_t^{(k)})$ denotes the state-action pair of the k th MDP, which is a pair of random variables determined by d_0 and $\mathcal{P}$. Define a constraint set

$$\mathcal{G} := \{(\mathcal{P}, d_0) : G_{i,T}(d_0, \mathcal{P}) \leq 0, i = 1, 2, \dots, m\}. \quad (2)$$

Define the regret of a policy $\mathcal{P}$ with respect to a particular joint randomized stationary policy Π along with an arbitrary starting state distribution d_0 as:

$$F_T(d_0, \mathcal{P}) - F_T(d_0, \Pi),$$

The goal of OCMDP is to choose a policy $\mathcal{P}$ so that both the regret and constraint violations grow sublinearly with respect to T , where regret is measured against all feasible joint randomized stationary policies Π . An important feature of this “weakly coupled” MDP structure is that, while the total state space $(s^{(1)}, \dots, s^{(K)})$ grows exponentially in the number of subsystems K , our solution can be implemented separately at each system $i \in \{1, \dots, K\}$ with complexity that depends only on the size of the individual system state $s^{(i)}$, rather than the product of sizes across all systems.

1.1 Related works

Online convex optimization (OCO): This concerns multi-round cost minimization with arbitrarily-varying convex loss functions. Specifically, on each slot t the decision maker chooses decisions $x(t)$ within a convex set $\mathcal{X}$ (before observing the loss function $f^t(x)$) in order to minimize the total *regret* compared to the best fixed decision in hindsight, expressed as:

$$\text{regret}(T) = \sum_{t=1}^T f^t(x(t)) - \min_{x \in \mathcal{X}} \sum_{t=1}^T f^t(x).$$

See [4] for an introduction to OCO. Zinkevich introduced OCO in [14] and shows that an online projection gradient descent (OGD) algorithm achieves $O(\sqrt{T})$ regret. This $O(\sqrt{T})$ regret is proven to be the best in [5], although improved performance is possible if all convex loss functions are *strongly* convex. The OGD decision requires to compute a projection of a vector onto a set $\mathcal{X}$. For complicated sets $\mathcal{X}$ with functional equality constraints, e.g., $\mathcal{X} = \{x \in \mathcal{X}_0 : g_k(x) \leq 0, k \in \{1, 2, \dots, m\}\}$, the projection can have high complexity. To circumvent the projection, work in [6, 7, 12] proposes alternative algorithms with simpler per-slot complexity and that satisfy the inequality constraints in the long term (rather than on every slot). Recently, new primal-dual type algorithms with low complexity are proposed in [8, 11] to solve more challenging OCO with time-varying functional inequality constraints. In particular, [11] also treats online convex optimization with stochastic i.i.d. constraints but without any Markov structure. Thus, in the degenerate scenario where there is only one state in the state-space $\mathcal{S}^{(k)}$, $\forall k$, our problem (1)-(2) can be solved via the method proposed in [11]. However, there is no prior work that addresses the general constrained online MDP problem.

Online Markov decision processes: This extends OCO to allow systems with a more complex Markov structure. This is similar to the setup of the current paper of minimizing the expression (1), but does not have the constraint set (2). Unlike traditional OCO, the current penalty depends not only on the current action and the current (unknown) penalty function, but on the current system state (which depends on the history of previous actions). Further, the number of policies can grow exponentially with the sizes of the state and action spaces, so that solutions can be computationally intensive. The work [2] develops an algorithm in this context with $O(\sqrt{T})$ regret. Extended algorithms and regularization methods are developed in [1][3][13] to reduce complexity and improve dependencies on the number of states and actions. Online MDP under bandit feedback (where the decision maker can only observe the penalty corresponding to the chosen action) is considered in [9][13].

2 OCMDP ALGORITHM

2.1 Preliminaries

Given an MDP with state space $\mathcal{S}$ and action space $\mathcal{A}$, a *policy* $\mathcal{P}$ defines a (possibly probabilistic) method of choosing actions $a \in \mathcal{A}$ at state $s \in \mathcal{S}$ based on the past information. A **randomized stationary policy** π defines an algorithm which, whenever the system is in state $s \in \mathcal{S}$, chooses an action $a \in \mathcal{A}$ according to a fixed conditional probability function $\pi(a|s)$, defined for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$. For any randomized stationary policy π , it holds that $\sum_{a \in \mathcal{A}^{(k)}} \pi(a|s) = 1$ for all $s \in \mathcal{S}^{(k)}$. Define the transition probability matrix $\mathbf{P}_\pi^{(k)}$ under policy π to have components as follows:

$$P_\pi^{(k)}(s, s') = \sum_{a \in \mathcal{A}^{(k)}} \pi(a|s) P_a^{(k)}(s, s'), \quad s, s' \in \mathcal{S}^{(k)}. \quad (3)$$

It is easy to verify that $\mathbf{P}_\pi^{(k)}$ is indeed a *stochastic matrix*, that is, it has rows with nonnegative components that sum to 1.

Next, we introduce the state-action polyhedron with respect to each MDP. Let $\Delta \subseteq \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be a probability simplex, i.e.

$$\Delta = \left\{ \theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} : \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \theta(s,a) = 1, \theta(s,a) \geq 0 \right\}.$$

Given a randomized stationary policy π with stationary state distribution d_π , the MDP is a Markov chain with transition matrix $\mathbf{P}_\pi$ given by (3). Thus, it must satisfy the following balance equation:

$$\sum_{s \in \mathcal{S}} d_\pi(s) P_\pi(s, s') = d_\pi(s'), \quad \forall s' \in \mathcal{S}.$$

Defining $\theta(a, s) = \pi(a|s) d_\pi(s)$ and substituting the definition of transition probability (3) into the above equation gives

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \theta(s, a) P_a(s, s') = \sum_{a \in \mathcal{A}} \theta(s', a), \quad \forall s' \in \mathcal{S}.$$

The variable $\theta(a, s)$ is often interpreted as a stationary probability of being at state $s \in \mathcal{S}$ and taking action $a \in \mathcal{A}$ under some randomized stationary policy. The state action polyhedron Θ is then defined as

$$\Theta := \left\{ \theta \in \Delta : \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \theta(s, a) P_a(s, s') = \sum_{a \in \mathcal{A}} \theta(s', a), \quad \forall s' \in \mathcal{S} \right\}.$$

Given any $\theta \in \Theta$, one can recover a randomized stationary policy π at any state $s \in \mathcal{S}$ as

$$\pi(a|s) = \begin{cases} \frac{\theta(a,s)}{\sum_{a \in \mathcal{A}} \theta(a,s)}, & \text{if } \sum_{a \in \mathcal{A}} \theta(a,s) \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

2.2 The proposed algorithm

Our proposed algorithm is distributed in the sense that each time slot, each MDP solves its own subproblem and the constraint violations are controlled by a simple update of global multipliers called “virtual queues” at the end of each slot. Let $\Theta^{(1)}, \Theta^{(2)}, \dots, \Theta^{(K)}$ be state-action polyhedron of K MDPs, respectively. Let $\theta_t^{(k)} \in \Theta^{(k)}$ be a state-action vector at time slot t . At $t = 0$, each MDP chooses its initial state-action vector $\theta_0^{(k)}$ resulting from any *separable* randomized stationary policy $\pi_0^{(k)}$. For example, one could choose a uniform policy $\pi_0^{(k)}(a|s) = 1/|\mathcal{A}^{(k)}|$, $\forall s \in \mathcal{S}^{(k)}$, solve the equation $d_{\pi_0^{(k)}} = d_{\pi_0^{(k)}} \mathbf{P}_{\pi_0^{(k)}}^{(k)}$ to get a probability vector $d_{\pi_0^{(k)}}$, and obtain $\theta_0^{(k)}(a,s) = d_{\pi_0^{(k)}}(s)/|\mathcal{A}^{(k)}|$. For each constraint $i \in \{1, 2, \dots, m\}$, let $Q_i(t)$ be a *virtual queue* defined over slots $t = 0, 1, 2, \dots$ with the initial condition $Q_i(0) = Q_i(1) = 0$, and update equation:

$$Q_i(t+1) = \max \left\{ Q_i(t) + \sum_{k=1}^K \left\langle \mathbf{g}_{i,t-1}^{(k)}, \theta_t^{(k)} \right\rangle, 0 \right\}. \quad (5)$$

Our algorithm uses two parameters $V > 0$ and $\alpha > 0$ and makes decisions as follows:

- At the start of each slot $t \in \{1, 2, 3, \dots\}$, the k -th MDP observes $Q_i(t)$, $i = 1, 2, \dots, m$ and chooses $\theta_t^{(k)}$ to solve the following subproblem:

$$\theta_t^{(k)} = \underset{\theta \in \Theta^{(k)}}{\operatorname{argmin}} \left\langle V \mathbf{f}_{t-1}^{(k)} + \sum_{i=1}^m Q_i(t) \mathbf{g}_{i,t-1}^{(k)}, \theta \right\rangle + \alpha \left\| \theta - \theta_{t-1}^{(k)} \right\|_2^2. \quad (6)$$

- Construct the randomized stationary policy $\pi_t^{(k)}$ according to (4) with $\theta = \theta_t^{(k)}$, and choose the action $a_t^{(k)}$ at k -th MDP according to the conditional distribution $\pi_t^{(k)}(\cdot|s_t^{(k)})$.
- Update the virtual queue $Q_i(t)$ according to (5) for all $i = 1, 2, \dots, m$.

Note that for any slot $t \geq 1$, this algorithm gives a *separable* randomized stationary policy, so that each MDP chooses its own policy based on its own function $\mathbf{f}_{t-1}^{(k)}, \mathbf{g}_{i,t-1}^{(k)}, i \in \{1, 2, \dots, m\}$, and a common multiplier $\mathbf{Q}(t) := (Q_1(t), \dots, Q_m(t))$.

2.3 Performance Theorem

Before stating the main theorem, we need the following definition which specify the set of benchmarking algorithms. A **joint randomized stationary policy** Π on K parallel MDPs defines an algorithm which chooses a joint action $\mathbf{a} := (a^{(1)}, a^{(2)}, \dots, a^{(K)}) \in \mathcal{A}^{(1)} \times \mathcal{A}^{(2)} \times \dots \times \mathcal{A}^{(K)}$ given the joint state $\mathbf{s} := (s^{(1)}, s^{(2)}, \dots, s^{(K)}) \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \dots \times \mathcal{S}^{(K)}$ according to a fixed conditional probability $\Pi(\mathbf{a}|\mathbf{s})$. The proof of the following theorem is given in [10].

THEOREM 2.1. *Let $\mathcal{P}$ be the sequence of randomized stationary policies resulting from the proposed algorithm with $V = \sqrt{T}$ and $\alpha = T$. Let d_0 be the starting state of the proposed algorithm. For any jointly randomized stationary policy Π starting from the state d_0 such that $(d_0, \Pi) \in \mathcal{G}$, we have*

$$F_T(d_0, \mathcal{P}) - F_T(d_0, \Pi) \leq O \left(m^{3/2} K^2 \sum_{k=1}^K |\mathcal{A}^{(k)}| |\mathcal{S}^{(k)}| \cdot \sqrt{T} \right),$$

$$G_{i,T}(d_0, \mathcal{P}) \leq O \left(m^{3/2} K^2 \sum_{k=1}^K |\mathcal{A}^{(k)}| |\mathcal{S}^{(k)}| \cdot \sqrt{T} \right), \quad i = 1, 2, \dots, m.$$

3 CONCLUSION AND FUTURE WORKS

This paper considers online learning over weakly coupled MDPs where the coupling comes from the global constraint functions, and the time varying objective and constraint functions can only be observed after the decision is made. We develop a new algorithm along with a new framework for analysis guaranteeing $O(\sqrt{T})$ regret and constraint violation simultaneously. Note that the current algorithm and analysis assume the full knowledge of the transition probabilities of underlying MDPs and the condition that the decision maker can observe the entire objective and constraint functions over all state-action pairs each slot after the decision is made. It would be interesting if one can relax the above assumptions, and develop algorithms with competitive regret bounds.

REFERENCES

- [1] Travis Dick, Andras Gyorgy, and Csaba Szepesvari. 2014. Online learning in Markov decision processes with changing cost sequences. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 512–520.
- [2] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. 2009. Online Markov decision processes. *Mathematics of Operations Research* 34, 3 (2009), 726–736.
- [3] Peng Guan, Maxim Raginsky, and Rebecca M Willett. 2014. Online Markov decision processes with Kullback-Leibler control cost. *IEEE Trans. Automat. Control* 59, 6 (2014), 1423–1438.
- [4] Elad Hazan et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization* 2, 3-4 (2016), 157–325.
- [5] Elad Hazan, Amit Agarwal, and Satyen Kale. 2007. Logarithmic regret algorithms for online convex optimization. *Machine Learning* 69 (2007), 169–192.
- [6] Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. 2016. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, 402–411.
- [7] Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. 2012. Trading regret for efficiency: online convex optimization with long term constraints. *Journal of Machine Learning Research* 13, Sep (2012), 2503–2528.
- [8] Michael J Neely and Hao Yu. 2017. Online Convex Optimization with Time-Varying Constraints. *arXiv preprint arXiv:1702.04783* (2017).
- [9] Gergely Neu, Andras Antos, András György, and Csaba Szepesvári. 2010. Online Markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems*, 1804–1812.
- [10] Xiaohan Wei, Hao Yu, and Michael J Neely. 2018. Online Learning in Weakly Coupled Markov Decision Processes: A Convergence Time Study. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (2018), 12.
- [11] Hao Yu, Michael Neely, and Xiaohan Wei. 2017. Online Convex Optimization with Stochastic Constraints. *Advances in Neural Information Processing Systems* (2017), 1427–1437.
- [12] Hao Yu and Michael J Neely. 2016. A Low Complexity Algorithm with $O(\sqrt{T})$ Regret and Finite Constraint Violations for Online Convex Optimization with Long Term Constraints. *arXiv preprint arXiv:1604.02218* (2016).
- [13] Jia Yuan Yu, Shie Mannor, and Nahum Shimkin. 2009. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research* 34, 3 (2009), 737–757.
- [14] Martin Zinkevich. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 928–936.