Bias-Corrected Q-Learning With Multistate Extension

Donghun Lee and Warren B. Powell, *Member, IEEE*

***Abstract*—Q-learning is a sample-based model-free algorithm that solves Markov decision problems asymptotically, but in finite time, it can perform poorly when random rewards and transitions result in large variance of value estimates. We pinpoint its cause to be the estimation bias due to the maximum operator in Q-learning algorithm, and present the evidence of max-operator bias in its Q value estimates. We then present an asymptotically optimal bias-correction strategy and construct an extension to bias-corrected Q-learning algorithm to multistate Markov decision processes, with asymptotic convergence properties as strong as those from Q-learning. We report the empirical performance of the bias-corrected Q-learning algorithm with multistate extension in two model problems: A multiarmed bandit version of Roulette and an electricity storage control simulation. The bias-corrected Q-learning algorithm with multistate extension is shown to control max-operator bias effectively, where the bias-resistance can be tuned predictably by adjusting a correction parameter.**

***Index Terms*—Bias correction, electricity storage, Q-learning, smart grid.**

I. Introduction

RANDOMNESS arises in a number of stochastic optimization problems, such as stochastic shortest path problems, asset valuation problems, energy storage problems, and equipment repair problems. Randomness in the rewards can introduce a high level of uncertainty in estimates of the value of being in a state, which can complicate online learning algorithms as well as Monte Carlo based offline learning algorithms, which use sampled estimates of value functions. Randomness in the state transition can introduce another dimension of uncertainty in the value estimation in which state space is used as one of the estimation parameters. In this paper, we identify serious problems that this behavior introduces into practical implementations of Q-learning, a reinforcement learning algorithm that is frequently chosen as a go-to method for solving online, model-free stochastic control problems due to its desirable theoretical guarantees.

Q-learning, first proposed in [1], is a model-free approximate algorithm that asymptotically produces a Bellman optimal solution and solves Markov decision process (MDP) problems as dynamic programming problems without explicit knowledge of the distribution of the uncertainty. Functionally, the algorithm can be seen as a blend of exact value iteration and stochastic approximation as follows:

$$\hat{Q}^n \leftarrow \hat{C}\left(s^n, a^n\right) + \gamma \max_{a' \in \mathcal{A}\left(s^{n+1}\right)} \left(\bar{Q}^{n-1}\left(s^{n+1}, a'\right)\right) \tag{1}$$

$$\bar{Q}^n\left(s^n, a^n\right) \leftarrow \left(1 - \alpha\left(s^n, a^n\right)\right) \bar{Q}^{n-1}\left(s^n, a^n\right) + \alpha\left(s^n, a^n\right) \hat{Q}^n \tag{2}$$

where $\left(s^n, a^n\right)$ is a determined state-action pair, and s^{n+1} is a realization of random state transition due to taking action a^n in state s^n (we defer the detailed definition of other terms). Also, when the $\bar{Q}^n$ estimate is represented in tabular format, Q-learning enjoys asymptotic convergence properties with a mild set of technical assumptions as demonstrated in [2]–[4]. The assumptions in [2] allow many stochastic models for $\hat{C}$ and s^{n+1} given s^n, a^n that can be applied to Q-learning with its convergence guarantee. Moreover, the asymptotic rate of convergence of Q-learning has been studied theoretically by a number of authors including [5]–[7]. Thanks to its generally applicable set of assumptions and robust theoretical properties, Q-learning has been applied to a wide range of dynamic programming problems, including soccer-playing robot control [8], [9], human-computer dialogue strategy [10], pricing in agent-based economy [11], mobile robot navigation [12], computer game AI development [13], agent-based production scheduling [14], signal transmission system control [15].

However, when the Q-value estimator $\bar{Q}^n$ contains large stochastic noise, which may be due to a noisy contribution function or a highly stochastic transition function, Q-learning may suffer from finite-time bias as noted in [16] and [17]. This bias may cause Q-learning to generate its value estimate $\bar{V}_Q^n(s) := \max_a \bar{Q}^n(s, a)$ that significantly overestimates the true Bellman-optimal value $V^*(s)$. This overestimation can linger even after millions of sample observations, in applications where randomness in the contribution function or the transition function is sufficiently large. Consequently, Q-learning may show very slow real-world convergence of $\bar{V}_Q^n$ to $V^*(s)$, which may misguide many practitioners to prematurely declare the convergence of Q-learning. Later, during an attempt to

Manuscript received July 27, 2018; revised November 13, 2018; accepted November 23, 2018. Date of publication April 22, 2019; date of current version September 25, 2019. Recommended by Associate Editor U. V. Shanbhag. *(Corresponding author: Donghun Lee.)*

D. Lee is with the Department of Computer Science, Princeton University, Princeton, NJ 08540 USA (e-mail: donghunl@princeton.edu).

W. B. Powell is with the Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08540 USA (e-mail: powell@princeton.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2019.2912443

extract insights to the underlying problem from the "converged" Q-learning algorithm, practitioners may find out that their value estimates are wildly missing the theoretical optimal. The underlying cause of this phenomenon is the bias due to the $\max$ operator inherent in Q-learning as shown in (1).

Our work is motivated by two different problem settings: The betting strategy in the game of Roulette, and the charge-discharge control algorithm of an electricity storage attached to a smart grid. The two problems share the characteristic of uncertainty in problem structure, although the source of uncertainty of the two problems is quite distinct. We also note that both have discount factors very close to 1. This creates a setting where the max-operator bias becomes severe. Yet, the purpose of applying algorithms like Q-learning is not only to find out the optimal control policy that obtains Bellman optimality, but also to find out the correct time-discounted value of running the control policy, since that value estimate *is* the estimate of actual fiscal value of the equivalent real-world item that enables the policy. These are the situations where overestimating the value *will* lead to unwise decisions.

In this paper, we present a significant extension of discussion on max-operator bias shown in [18], through characterizing the max-operator bias into two types by the fundamental sources of the bias in value estimates of Q-learning. We also construct additive correction term that addresses both sources of max-operator bias, and use it to present bias-corrected Q-learning with multistate extension (BCQ-MS). Also, the bias-correction term originally reported in [19] is modified for robustness in the form of bias-corrected Q-learning we present in this paper. We prove the asymptotic unbiasedness of the bias correction and the asymptotic convergence of the BCQ-MS algorithm. We present the effect of bias correction by comparing the value estimates of classical Q-learning and those of BCQ-MS.

The rest of the paper is organized as follows. First, Section II characterizes the max-operator bias in Q-learning. Section III derives the bias correction factor for a single-state MDP (SS-MDP), with which we also construct a more robust version of the bias-corrected Q-learning algorithm for SS-MDP and demonstrate the $|\mathcal{A}|$-asymptotic unbiasedness of this correction. Section IV is where we present the bias-corrected Q-learning (BCQ) algorithm with multistate extension and its asymptotic convergence to Bellman optimal value function. We provide empirical evidence of bias-corrected Q-learning and its multistate extension showing effective mitigation of the max-operator bias found in Q-learning applied to the roulette benchmark problem in Section V and to the battery control benchmark problem in Section VI.

II. Characterization of Max-Operator Bias in Q-learning

We first show an example of max-operator bias in Q-learning. We define the terms to provide an intuitive glimpse of the source of max-operator bias in Q-learning, and then construct an oracle Q-learning process. Using the oracle process, we define the max-operator bias and characterize its existence condition.

A. Structural Consequence of Q-learning: Max-Operator Bias

Q-learning is an iterative algorithm that solves an MDP defined by the 5-tuple $(\mathcal{S}, \mathcal{A}, C, T, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $C : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the contribution function, $T : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the transition function, and $\gamma \in [0, 1)$ is the discount factor.

Running a Q-learning algorithm or a variant that follows the key update equations (1) and (2), under an MDP defined by the 5-tuple $(\mathcal{S}, \mathcal{A}, C, T, \gamma)$, will generate $\{(s^n, a^n, \hat{C}^{n+1}, \hat{T}^{n+1})\}_{n=0,1,\ldots}$, a sequence of state, action, observed contribution, and observed transition.

Using the iteration counter n to denote time, we index the current state as s^n, the action as a^n, the observed contribution as $\hat{C}^{n+1} \sim C(s^n, a^n)$, and the observed transition to the next state (s^{n+1}) as $\hat{T}^n \sim T(s^n, a^n)$. We also denote a probability space $(\Omega, \mathfrak{F}, \mathbb{P})$ where all possible trajectories that the Q-learning algorithm can take are in Ω, and the filtration $\mathfrak{F}$ is the limiting σ-algebra of the filtration $\mathfrak{F}^0 \subseteq \mathfrak{F}^1 \subseteq \mathfrak{F}^2 \subseteq \cdots$ on Ω indexed by $n = 0, 1, 2, \ldots$. Each filtration $\mathfrak{F}^n$ is a σ-algebra generated by an increasing sequence of random variables $(s^0, a^0, \hat{C}^1, \hat{T}^1, \ldots, \hat{C}^n, \hat{T}^n)$.

With an appropriately chosen stepsize rule $\alpha(s, a)$ and an exploration policy that allows sufficient coverage of $(s, a) \in \mathcal{S} \times \mathcal{A}$, Q-learning generates output $\bar{Q}^n$ that, as $n \to \infty$, asymptotically approaches the solution $\bar{Q}^*$ that satisfies Bellman optimality as

$$\bar{Q}^*(s, a) = C(s, a) + \gamma \max_{a' \in \mathcal{A}} \left\{ \bar{Q}^*(T(s, a), a') \right\} \tag{3}$$

for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. This asymptotic convergence still holds even when $C(s, a)$ and $T(s, a)$ are random variables. The optimality condition holds in conditional expectations as follows:

$$\begin{aligned} \bar{Q}^*(s, a) = {} & \mathbb{E}[C(s, a) | s, a] \\ & + \mathbb{E}\left[\gamma \max_{a' \in \mathcal{A}} \left\{ \bar{Q}^*(T(s, a), a') \right\} \middle| s, a \right] \end{aligned} \tag{4}$$

where the conditioning event s, a is a shorthand such that $\mathbb{E}[\cdot | s, a] := \mathbb{E}[\cdot | S = s, A = a]$, in which S, A are random variables that take values in $\mathcal{S}, \mathcal{A}$, respectively.

problems may generate output $\bar{Q}^n$ with a significant deviation from the asymptotic optimal value $\bar{Q}^*$. We name the cause of this phenomenon in Q-learning as "max-operator bias," as the $\max$ operator in Q-learning algorithm plays a central role in manifesting the deviation in $\bar{Q}^n$. We first provide an intuitive explanation on how $\max$ operator is involved in this. As Q-learning estimate $\bar{Q}^n$ is a stochastic approximation of Bellman optimal value $\bar{Q}^*$, its deviation originates from the deviations in sample realizations $\hat{C}$ and $\hat{T}$. The deviation, when positive, can be preserved and propagated by the update formula used by Q-learning as outlined below. Let us assume such positive deviation happened at iteration m, where $\hat{C}^{m+1} > \mathbb{E}C(s^m, a^m)$, and $\hat{T}^{m+1}$ was chosen by chance such that $\gamma \max_{a'} \bar{Q}^{m-1}(s^{m+1}, a')$ was sufficiently large to have positive bias in $\hat{Q}^m$. This bias is transferred to $\bar{Q}^m$ as shown in (2), and let us assume that this bias is largest such that $\bar{Q}^m(s^m, a^m) = \max_{a'} \bar{Q}^m(s^m, a')$. Then,

later at iteration $n > m$ where $\hat{T}^{n+1} := T(s^n, a^n)$ happened to be $s^{n+1} = s^m$, the positive bias in $\bar{Q}^m$ is now propagated to $\bar{Q}^n$ due to the formula in (1). A similar procedure can be repeated with different stochastic deviations that depend on the variance of C and T. As explained above, the max operator found in Q-learning update rule in (1) is the key operator that asymmetrically preserves only positive deviations and allows their propagation.

Before analyzing the deviation of $\hat{Q}^n$ and bias in $\bar{Q}^n$ in any finite iteration n, we introduce a conditional expectation shorthand $\mathbb{E}^n[\cdot]$. We use this as the conditional expectation with respect to sigma-algebra $\mathfrak{F}^n$, which means taking the expectation conditioned on all information available at iteration n, including all the observed random variables indexed by up to n. In the context of Q-learning applied to an MDP, we note that the observed transition $\hat{T}^n$ is s^n, because this transition is caused by taking action a^{n-1} from state s^{n-1} in the given MDP. Therefore, the shorthand $\mathbb{E}^n$ can be explicitly interpreted as

$$\begin{aligned}\mathbb{E}^n[\cdot] &:= \mathbb{E}[\cdot|\mathfrak{F}^n] \\ &= \mathbb{E}[\cdot|s^0, a^0, \hat{C}^1, \ldots, s^{n-1}, a^{n-1}, \hat{C}^n, \hat{T}^n] \\ &= \mathbb{E}[\cdot|s^0, a^0, \hat{C}^1, \ldots, s^{n-1}, a^{n-1}, \hat{C}^n, s^n] \\ &= \mathbb{E}[\cdot|s^0, a^0, \hat{C}^1, \ldots, s^{n-1}, a^{n-1}, \hat{C}^n, s^n, a^n]. \end{aligned} \quad (5)$$

Note that we can arbitrarily include a^n among the conditioned variables, since a^n is a $\mathfrak{F}^n$-measurable random variable. In a similar manner, we shorthand the conditional variance with respect to $\mathfrak{F}^n$ as $\mathrm{Var}^n[\cdot]$.

Now we are ready to define the bias materialized by max operator from stochastic sample deviations in C and T.

Definition 1 (Max-operator bias in Q-learning): The max-operator bias of Q-learning sample $\hat{Q}^n$, at iteration counter n is defined as

$$B^n := \hat{Q}^n - \mathbb{E}[\hat{Q}^n|\mathfrak{F}^{n-1}]. \quad (6)$$

This is the actual bias observed at iteration n, when the sample $\hat{Q}^n$ is observed. Under certain conditions, this bias will accumulate significantly in Q-learning output $\bar{Q}^n(s, a)$, which may lead to an unreasonably large estimate of the value function $\bar{V}^n(s) := \max_{a \in \mathcal{A}(s)} \bar{Q}^n(s, a)$.

At the first glance, the max operator in (1) is the source of nonnegative bias to $\hat{Q}^n$ as noted in [16] and [17], for the maximum of the random variables $\bar{Q}^{n-1}$ is on average no less than the average of $\bar{Q}^{n-1}$ itself. This max-operator bias is propagated through the value estimates $\bar{Q}^n$ by stochastic approximation step in (2). The max-operator bias is inherent in the Q-learning algorithm, but it can be damped out, for example, as the discount factor γ approaches 0.

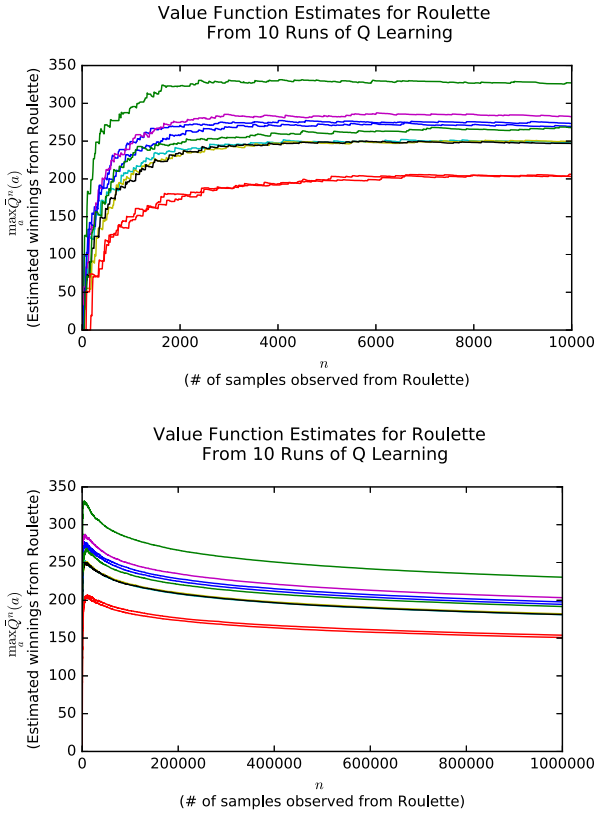

Fig. 1. Showcase of overestimated value functions from Q-learning. Each figure has ten lines corresponding each sample runs. Each line shows the evolution of Q value $\bar{Q}^n$ versus n. The first figure is showing the first 1% of the iterations (up to 100 K steps) that are shown in the second figure (which shows up to 1 M steps). The correct value estimate is 0.

B. Empirical Evidence

We provide empirical evidence of the overestimation of the value function caused by the accumulation of max-operator bias in Q-learning. The experimental setting is a simplified American roulette, where the action set is comprised of betting \$1 at one of 38 numbers, plus not participating in the bet. It is well known that the act of participating in this particular gamble has negative average expected return. We know that the best action in front of a perfect roulette table is to not participate, and the optimal value function is, therefore, 0.

The roulette MDP is set up as a single-state, 39-action, $\gamma = 0.99$ infinite-horizon discounted MDP with a reward function of 35 upon correct bet, -1 on wrong bet, and 0 on not placing any bet. We used pure exploration policy in choosing action a^n instead of greedy policy in which the past experience affects the distribution of upcoming actions, since our goal is to demonstrate the significance of max-operator bias in Q-learning even under most conservative exploration policy. We repeat the algorithm with ten different random seeds.

As shown in Fig. 1, there is a consistent overestimation of the estimated value function from Q-learning (y axis) of the true value function (constant 0). The overestimation may be easily mistaken as indicating convergence since the estimates

self-correct only very slowly after the initial transient phase. Yet, even after observing 10 00 000 roulette rolls, the value estimates from Q-learning are still far from the true value 0, as shown in the second subfigure. Any premature declaration of convergence may result in a misleading policy such that according to the policy, playing the game is expected to produce positive rewards. The problem of max-operator bias in Q-learning value estimate $\bar{Q}^n$ is real and can persist for many iterations.

C. Oracle Q-learning Process and its Properties

We introduce the oracle Q-learning process as a tool to facilitate analysis of the max-operator bias in Q-learning. Oracle Q-learning is the dream of ordinary Q-learning, in the sense that oracle Q-learning knows the contribution function C and the transition function T. The iterative update formula of Oracle Q-learning is, therefore, very similar to that of Q-learning, but with the ability to calculate the conditional expectation. We use superscript * to denote oracle Q-learning analogs as follows.

Definition 2 (Oracle Q Sample and Estimate): At iteration n of Q-learning, the oracle Q sample $\hat{Q}^{*,n}$ is defined as follows:

$$\hat{Q}^{*,n} \leftarrow \mathbb{E}[\hat{C}(s^n,a^n)|\mathfrak{F}^{n-1}] + \gamma\mathbb{E}\left[\max_{a'\in\mathcal{A}(S^{n+1})}\bar{Q}^{n-1}\left(S^{n+1},a'\right)\middle|\mathfrak{F}^{n-1}\right] \tag{7}$$

and the oracle Q estimates $\bar{Q}^{*,n}$ are updated as follows:

$$\bar{Q}^{*,n}(s^n,a^n) \leftarrow (1-\alpha(s^n,a^n))\bar{Q}^{n-1}(s^n,a^n) + \alpha(s^n,a^n)\hat{Q}^{*,n}. \tag{8}$$

The access to the oracle allows the exact computation of the expectations over $\hat{C}$ and $S^{n+1}=\hat{T}^n$ in (7). The oracle Q sample $\hat{Q}^{*,n}$ is in fact the unbiased sample with zero max-operator bias. It is straightforward to show this by substituting $\hat{Q}^n$ with $\hat{Q}^{*,n}$ in Definition 1. With oracle Q sample, we can interpret intuitively the max-operator bias as the difference between Q-learning sample $\hat{Q}^n$ and the oracle Q sample $\hat{Q}^{*,n}$, as $B^n := \hat{Q}^n - \mathbb{E}[\hat{Q}^n|\mathfrak{F}^{n-1}] = \hat{Q}^n - \hat{Q}^{*,n}$.

With the definitions and an intuitive understanding of the max-operator bias, we are ready to present the characteristics of max-operator bias in Q-learning.

D. Characterization and Existence of Max-Operator Bias in Q-learning

The max-bias term can be classified into two different portions. From its definition, we rearrange the terms as follows:

$$B^n := \hat{Q}^n - \mathbb{E}[\hat{Q}^n|\mathfrak{F}^{n-1}] \tag{9}$$

$$= \hat{C}(s^n,a^n) + \gamma\max_{a'\in\mathcal{A}(s^{n+1})}\bar{Q}^{n-1}\left(s^{n+1},a'\right) - \left(\mathbb{E}[\hat{C}(s^n,a^n)|\mathfrak{F}^{n-1}] + \gamma\mathbb{E}\left[\max_{a'\in\mathcal{A}(T^n)}\bar{Q}^{n-1}\left(\hat{T}^n,a'\right)\middle|\mathfrak{F}^{n-1}\right]\right) \tag{10}$$

$$= B_C^n + B_T^n \tag{11}$$

to isolate the two bias terms B_C^n and B_T^n. It is clear that there are two different sources of max-operator bias in Q-learning: One from the stochasticity of $\hat{C}$, the other from the stochasticity of $\hat{T}$. For clarity's sake, the expressions of the two terms are shown as follows:

$$B_C^n := \hat{C}(s^n,a^n) - \mathbb{E}^{n-1}\left[\hat{C}(s^n,a^n)\right] \tag{12}$$

$$B_T^n := \gamma\left(\max_{a'\in\mathcal{A}(s^{n+1})}\bar{Q}^{n-1}\left(s^{n+1},a'\right) - \mathbb{E}^{n-1}\left[\max_{a'\in\mathcal{A}(\hat{T}^n)}\bar{Q}^{n-1}\left(\hat{T}^n,a'\right)\right]\right). \tag{13}$$

Trivially, the bias due to stochasticity of $\hat{C}(s^n,a^n)$ disappears when $\mathrm{Var}^{n-1}\hat{C}(s^n,a^n)=0$, and the bias due to stochasticity of $\hat{T}^n$ disappears when either $\gamma=0$ or $|\mathcal{S}|=1$. However, unless both the conditions are met, B^n does not disappear in Q-learning even as $n\to\infty$. Instead, there is always a significant probability of $B^n>0$, which translates to the upward bias introduced into $\bar{Q}^n$. This upward bias contributes to the upward bias in $\hat{Q}^m$, the sample in the future iteration m in which the Q-learning revisits the state-action pair (s^n,a^n), because such positive bias B^n is reflected on $\bar{Q}^n$ and propagated to later Q-samples $\hat{Q}^m$ where $m>n$ and $(s^m,a^m)=(s^n,a^n)$. We show that there is nonzero probability of $B^n>0$ for all n.

Lemma 3: Given a Q-learning instance solving an MDP with at least one of $\hat{C}^n$ and $\gamma\max_{a'}\bar{Q}^{n-1}(\hat{T}^n,a')$ is a random variable, and the variables $\hat{C}^n$ and $\hat{T}^n$ are independent conditional on (s^n,a^n), max-bias exists in $\hat{Q}^n$, such that for all n

$$\mathbb{P}[B^n>0|\mathfrak{F}^{n-1}]>0. \tag{14}$$

Proof: We define $\hat{Y}^n := \gamma\max_{a'}\bar{Q}^{n-1}(\hat{T}^n,a')$. The conditional independence of $\hat{C}^n := C(s^n,a^n)$ and $\hat{T}^n = T(s^n,a^n)$ given (s^n,a^n) is true according to the definition of MDP, and this implies $\hat{C}^n$ and $\hat{Y}^n$ are independent.

First, we consider the case of both $\hat{C}^n$ and $\hat{Y}^n$ are random variables. When $\hat{C}^n$ and $\hat{Y}^n$ are random variables, $\hat{Q}^n=\hat{C}^n+\hat{Y}^n$ is a sum of two independent random variable with a pdf equal to the convolution of the pdf of $\hat{C}^n$ and $\hat{Y}^n$. Otherwise, when only one of the $\hat{C}^n$ and $\hat{Y}^n$ is a random variable, then the pdf of $\hat{Q}^n$ is a pdf of the underlying random variable, shifted by the value of the other deterministic variable. Therefore $\hat{Q}^n$ is another random variable given (s^n,a^n), so $\mathbb{P}[B^n>0|\mathfrak{F}^{n-1}] = \mathbb{P}[\hat{Q}^n-\mathbb{E}^{n-1}[\hat{Q}^n]>0|\mathfrak{F}^{n-1}]>0$. ■

As shown, at any iteration while a Q-learning algorithm is running, it is possible to have positive max-operator bias in a Q-learning sample $\hat{Q}^n$ in a generic MDP problem with very mild assumptions. Once the bias happens, it will influence Q estimate $\bar{Q}^n$ by the update rule in (2), and may have more effect in later iterations. This applies to ideal situations even when the Q estimate is exactly the true value as $\bar{Q}^n=\bar{Q}^*$.

The characterization of max-operator bias in Q-learning implies an immediate corollary for the sufficient condition when the max-operator bias cannot exist in Q-learning.

Corollary 4 (Condition for zero max-operator bias): The max-operator bias does not exist when the following conditions are both true.

1) $\mathrm{Var}^{n-1}\hat{C}^n = 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.
2) $\gamma = 0$ or $|\mathcal{S}| = 1$.

Proof: The first condition removes the B_C^n term of the bias, by removing the randomness of $\hat{C}(s^n, a^n)$. The second condition removes the B_T^n term of the bias, by removing the randomness of $\gamma \max_{a'} \bar{Q}^{n-1}(\hat{T}^n, a')$. ■

We wish to create a simple problem class, which still exhibits max-operator bias. The goal is to use this simple stylized MDP problem to derive the bias term analytically and use it as the basis of a bias correction term. Roulette is a nice example of this class of problem, which we call the SS-MDP.

Definition 5 (SS-MDP): An SS-MDP in this paper is an MDP-tuple $(\mathcal{S}, \mathcal{A}, C, T, \gamma)$ such that $|\mathcal{S}| = 1$, $|\mathcal{A}| > 1$, $\forall (s,a) \in \mathcal{S} \times \mathcal{A} : \mathrm{Var} C(s,a) > 0$, and $\gamma \neq 0$.

III. Bias-Correction for SS-MDP

We introduce the desired property of an ideal max-operator bias correction term for Q-learning in SS-MDP. Then we present the bias correction term $\tilde{B}_C^{n-1}$ for SS-MDP and its incorporation into Q-learning algorithm as BCQ for SS-MDP. Then we show the structural property of BCQ for SS-MDP that the term $\bar{B}_C^{n-1}$ reaches asymptotically unbiased correction as $|\mathcal{A}| \to \infty$.

A. Bias Correction Term

We first present the desired property of a max-operator bias correction term.

Proposition 6 (Unbiasedness Condition for Max-operator Bias Correction): Let $\hat{Q}^n$ be Q-sample value from observing random variables in Q-learning iteration $n < \infty$. A bias correction term $\bar{B}^{n-1}$ is unbiased when the expectation of bias-corrected sample $\hat{Q}^n - \bar{B}^{n-1}$ satisfies

$$\mathbb{E}[\hat{Q}^n - \bar{B}^{n-1} | \mathfrak{F}^{n-1}] - \hat{Q}^{*,n} = 0 \tag{15}$$

where $\hat{Q}^{*,n}$ denotes oracle Q-sample defined as

$$\hat{Q}^{*,n} := \mathbb{E}[\hat{C}\left(s^{n-1}, a^{n-1}\right) | \mathfrak{F}^{n-1}] + \gamma \mathbb{E}\left[\max_{a \in \mathcal{A}(\hat{T}^n)} \bar{Q}^{n-1}\left(\hat{T}\left(s^{n-1}, a^{n-1}\right)\right) \middle| \mathfrak{F}^{n-1}\right]. \tag{16}$$

By definition, oracle Q-sample is bias free, as it uses not the sample realization, but the expectation of the random variables. However, we do not assume access to Oracle Q sample to reach the desired property of Proposition 6. For SS-MDP, we present a $\mathfrak{F}^n$-measurable function $\tilde{B}_C^n$ as a bias correction term, as B_C^n is the only source of bias and $B_T^n = 0$ in SS-MDP.

Definition 7: A bias correction term $\tilde{B}_C^n$ is defined as

$$\tilde{B}_C^n(s^n, a^n) = \left(\frac{\xi}{b_{|\mathcal{A}(s^n)|}} + b_{|\mathcal{A}(s^n)|}\right) \sigma(s^n, a^n) \tag{17}$$

Algorithm 1: BCQ for SS-MDP.

Require: $\bar{Q}^0(s,a)$, s^0, γ, access to MDP $(\mathcal{S}, \mathcal{A}, C, T)$, N, stepsize rule α_n

1: **for** $n = 0, 1, \ldots, N-1$ **do**
2: Decide a^n
3: Observe $\hat{C}^{n+1} \sim C(s^n, a^n)$
4: Observe $s^{n+1} = \hat{T}^{n+1} \sim T(s^n, a^n)$
5: **if** $\tilde{B}_C^{n+1}$ is computable **then**
6: Compute $\tilde{B}_C^{n+1} \leftarrow \left(\frac{\xi}{b_{|\mathcal{A}(s^n)|}} + b_{|\mathcal{A}(s^n)|}\right) \bar{\sigma}_{n+1}(s^n, a^n)$
7: Compute $\hat{Q}_{BC}^{n+1} \leftarrow \bar{C}^{n+1}(s^n, a^n) + \gamma \max_{a \in \mathcal{A}(s^{n+1})} \bar{Q}^n\left(s^{n+1}, a\right) - \tilde{B}_C^{n+1}$
8: Update $\bar{Q}^{n+1}(s^n, a^n) \leftarrow (1 - \alpha_n(s^n, a^n))\bar{Q}^n(s^n, a^n) + \alpha_n(s^n, a^n)\hat{Q}_{BC}^{n+1}$
9: **end if**
10: **end for**
11: **return** $\bar{Q}^N(s,a)$

where ξ is Euler–Mascheroni constant and b_M is defined for $m > 0$ as

$$b_M := (2\log(M+7) - \log\log(M+7) - \log 4\pi)^{\frac{1}{2}} \tag{18}$$

and $\sigma(s,a) := \sqrt{\mathrm{Var}[C(s,a)]}$.

In problems where the variance of $C(s,a)$ is not known, $\sigma(s,a)$ is replaced by the square root of sample variance estimator of $C(s,a)$ that is computed with samples $\hat{C}^i$ observed for $i < n$.

B. Bias-Corrected Q-Learning Algorithm

The BCQ algorithm is shown in Algorithm 1, where we apply the concept of bias correction for SS-MDP to a conventional Q-learning algorithm. The key property of NCQ algorithm is the bias correction term $\tilde{B}_C^n$ which is incorporated into computing the $\hat{Q}_{BC}^n$.

Note that the condition for "$\tilde{B}_C^n$ is computable," as mentioned in Algorithm 1, is a gatekeeper to remind that the algorithm needs at least two samples of $\hat{C}(s,a)$ to compute $\tilde{B}_C^n(s,a)$ when the variance of $C(s,a)$ is not known. We denote the square root of sample variance estimator as $\bar{\sigma}_n(s,a)$ in line 6 of Algorithm 1. $\bar{C}^n(s,a)$, used in line 7 of Algorithm 1 is the sample mean of $C(s,a)$ using information up to iteration n, computed as

$$\bar{C}^n(s,a) := \frac{1}{|\mathcal{N}^n(s,a)|} \sum_{i \in \mathcal{N}^n(s,a)} \hat{C}\left(s^i, a^i\right) \tag{19}$$

where $\mathcal{N}^n(s,a)$ contains all time index $0 \leq i < n$ where (s,a) state-action pair was taken at index i. When there are not enough samples of $\hat{C}$, the algorithm defers updating the $\bar{Q}$ values. It is possible to use a more conservative condition by waiting for more than two samples for more stable estimates, especially when the uncertainty of C is too large to have a reasonable estimate with just two samples. This burn-in sample size is a tunable parameter, increasing which will make the correction term less

susceptible to deviations at the cost of delaying the onset of the learning. We use the minimum burn-in size of two samples throughout the experimental results presented in this paper.

Similar to Q-learning, it is important to select actions a^n to visit each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ infinitely often as $n \to \infty$ to ensure the limit value of $\bar{Q}^n$ to satisfy Bellman optimality.

C. Mathematical Preliminaries

We first prove several lemmas regarding b_M, the expression found in the correction term for B_C^n, and then other lemmas regarding asymptotic convergence in distribution of the maximum of normally distributed random variables to Gumbel random variables using b_M as a transformation formula. The lemmas introduced here are used to show the asymptotic unbiasedness property of the max-operator bias correction term $\tilde{B}_C^{n-1}$ in the BCQ algorithm for SS-MDP.

Lemma 8: $\forall u \in [x, \infty)$, $\frac{u^2}{2b_M^2} \xrightarrow{M\to\infty} 0$ where b_M is defined as in (18).

Proof: Substituting in the definition of b_M gives

$$\lim_{M\to\infty} \frac{u^2}{2b_M^2} = \lim_{M\to\infty} \frac{u^2}{O(\log(M+7))} = 0.$$

■

Lemma 9: $\frac{1}{2}b_M^2 + \log b_M - \log M + \frac{1}{2}\log 2\pi \xrightarrow{M\to\infty} 0$.

Proof: Substituting the definition of b_M (18) for b_M^2 part gives

$$\begin{aligned}
&\frac{1}{2}b_M^2 + \log b_M - \log M + \frac{1}{2}\log 2\pi \\
&= \frac{1}{2}\left(2\log(M+7) - \log\log(M+7) - \log 4\pi\right) \\
&\quad + \log b_M - \log M + \frac{1}{2}\log 2\pi \\
&= \log\left(\frac{M+7}{M}\right) + \log\left(\frac{b_M}{\sqrt{2\log M}}\right) \xrightarrow{M\to\infty} 0 + 0 = 0.
\end{aligned}$$

The last line mentioned above holds because $\log$ is a monotone function and $\frac{b_M}{\sqrt{2\log M}} \xrightarrow{M\to\infty} 1$ (l'Hôpital's rule). ■

Lemma 10: Given a normal c.d.f. $\Phi(x)$

$$\frac{-\log\Phi\left(\frac{x}{b_M} + b_M\right)}{1 - \Phi\left(\frac{x}{b_M} + b_M\right)} \xrightarrow{M\to\infty} 1.$$

Proof: Apply l'Hôpital's rule to obtain

$$\lim_{M\to\infty} \frac{-\log\Phi\left(\frac{x}{b_M} + b_M\right)}{1 - \Phi\left(\frac{x}{b_M} + b_M\right)} = \lim_{M\to\infty} \frac{-\frac{1}{\Phi\left(\frac{x}{b_M} + b_M\right)}}{-1} = 1.$$

The required conditions to apply l'Hôpital's rule are checked as follows.

1) $\lim_{M\to\infty}\left(-\log\Phi\left(\frac{x}{b_M} + b_M\right)\right) = 0$.
2) $\lim_{M\to\infty}\left(1 - \Phi\left(\frac{x}{b_M} + b_M\right)\right) = 0$.
3) Both functions are differentiable over the domain of M.

■

Lemma 11: Given an independent and identically distributed normal sample $\hat{X}_1, \hat{X}_2, \ldots \hat{X}_M$

$$b_M \cdot \left(\max_i \left\{ \frac{\hat{X}_i - \mathbb{E}\hat{X}_i}{\sqrt{\mathrm{Var}\hat{X}_i}} \middle| i \in \{1,2,\ldots,M\} \right\} - b_M \right) \xrightarrow[M\to\infty]{d} \mathcal{G}(0,1)$$

where $\mathcal{G}(0,1)$ is a standard Gumbel distribution which has mean $\xi \approx 0.5774$ (the Euler–Mascheroni constant) and variance $\frac{\pi^2}{6}$.

Proof: To show the convergence in distribution, we show the c.d.f. of the maximum of M standard normal random variables converges to the c.d.f. of Gumbel distribution. First, we use Φ, the c.d.f. of a single standard normal, to rewrite the c.d.f. of the maximum of M standard normal random variables as follows:

$$\begin{aligned}
&\Pr\left[\max\left(X_1,\ldots,X_M\right) \le x\right] \\
&= \Pr\left[\{X_1 \le x\} \cap \{X_2 \le x\} \cap \cdots \cap \{X_M \le x\}\right] \\
&= \prod_{i=1}^{M} \Pr\left[X_i \le x\right] \\
&= \left(\Phi(x)\right)^M.
\end{aligned}$$

Since $\hat{X}_1, \hat{X}_2, \ldots \hat{X}_M$ are normally distributed, $\frac{\hat{X}_i - \mathbb{E}\hat{X}_i}{\sqrt{\mathrm{Var}\hat{X}_i}}$ follows the standard normal distribution. So we prove the lemma by showing that

$$\left(\Phi\left(\frac{x}{b_M} + b_M\right)\right)^M \xrightarrow{M\to\infty} e^{-e^{-x}} =: G(x) \tag{20}$$

since the LHS is equivalent to $\Pr\left[\max\{X_i\} \le x\right]$ where X_i's are i.i.d. standard normal samples, and RHS is the c.d.f. of the standard Gumbel distribution.

The relation in (20) is equivalently stated as

$$\begin{aligned}
&\lim_{M\to\infty}\left(\Phi\left(\frac{x}{b_M} + b_M\right)\right)^M = G(x) \\
&\lim_{M\to\infty} M\log\left(\Phi\left(\frac{x}{b_M} + b_M\right)\right) = \log G(x).
\end{aligned}$$

Then it can be further transformed using Lemma 10

$$\begin{aligned}
&\lim_{M\to\infty} M\left(-\left(1 - \Phi\left(\frac{x}{b_M} + b_M\right)\right)\right) = \log G(x) \\
&\lim_{M\to\infty} M\left(1 - \Phi\left(\frac{x}{b_M} + b_M\right)\right) = -\log G(x).
\end{aligned} \tag{21}$$

Taking only the portion inside $\lim$ in the LHS of (21), we proceed as

$$
\begin{aligned}
& M\left(1-\Phi\left(\frac{x}{b_M}+b_M\right)\right) \\
&= \frac{M}{b_M\sqrt{2\pi}}\int_x^\infty \exp\left(\frac{-1}{2}\left(\frac{u^2}{b_M^2}+2u+b_M^2\right)\right)du \\
&= \exp\left(-\left(\frac{1}{2}b_M^2+\log b_M-\log M+\frac{1}{2}\log 2\pi\right)\right) \\
&\quad \int_x^\infty \exp\left(-\frac{u^2}{2b_M^2}\right)\exp\left(-u\right)du \\
&\xrightarrow{M\to\infty} 1\int_x^\infty 1\exp\left(-u\right)du \qquad (22)\\
&= e^{-x} \qquad (23)
\end{aligned}
$$

where (22) is true by Lemmas 8 and 9.

Since the c.d.f. of the standard Gumbel distribution is $G(x) := e^{-e^{-x}}$, so is $-\log G(x) = e^{-x}$. Therefore, (23) proves (21), proves (20) and the lemma. ■

D. Asymptotically Unbiased Correction for SS-MDP

Here we show that the bias correction term in Definition 7 asymptotically converges to the unbiased correction term as $|\mathcal{A}| \to \infty$ in SS-MDP. We also assume that contribution has a finite bound $|C| < C^M < \infty$.

Theorem 12: Given n is large enough to assume $\bar{C}_a^n$ for $a \in \mathcal{A}$ are i.i.d. normally distributed, the bias correction term $\tilde{B}_C^n$ converges to an unbiased correction term $\bar{B}^{*,n}$ as $|\mathcal{A}| \to \infty$ in SS-MDP.

Proof: For bias-corrected Q-sample $\hat{Q}_{BC}^n$ that includes the correction term $\tilde{B}_C^n$ to be unbiased asymptotically, it must satisfy the unbiasedness condition (as in Definition 6) asymptotically as follows:

$$
\mathbb{E}[\hat{Q}_{BC}^n|\mathfrak{F}^{n-1}] - \hat{Q}^{*,n} \xrightarrow{|\mathcal{A}|\to\infty} 0 \qquad (24)
$$

where $\hat{Q}_{BC}^n$ is computed as shown in line 7 of Algorithm 1. We show the $|\mathcal{A}|$-asymptotic unbiasedness of bias correction term $\tilde{B}_C^n$ by showing (24) in SS-MDP. As $|\mathcal{S}| = 1$ for SS-MDP, for brevity we omit s when it is used as an argument, for example using (a) instead of (s,a).

We assume at iteration n, a greedy action a^n is chosen such that $a^n = \arg\max_{a\in\mathcal{A}} \bar{C}^n(a)$. For succinctness, we use the following shorthand notations: The sample estimate $\bar{C}_a^n := \bar{C}^n(a)$, and $\bar{Q}_{\mathcal{M}}^n := \max_{a\in\mathcal{A}} \bar{Q}^n(a)$. We define $a^{*,n} := \arg\max_{a\in\mathcal{A}} \mathbb{E}^{n-1}\hat{Q}^n(a)$ as the optimal greedy action that maximizes the expectation of $\hat{Q}^n$ conditional to $\mathfrak{F}^{n-1}$. This notation allows us to rewrite $\hat{Q}^{*,n} = \mathbb{E}\bar{C}_{a^{*,n}} + \gamma\bar{Q}_{\mathcal{M}}^{n-1}$. Substituting $\hat{Q}^{*,n}$ into the LHS of (24) gives

$$
\begin{aligned}
& \mathbb{E}\left[\hat{Q}_{BC}^n - \tilde{B}_C^n \middle| \mathfrak{F}^{n-1}\right] - \hat{Q}^{*,n} \\
&= \mathbb{E}^{n-1}\left[\max_{a\in\mathcal{A}} \bar{C}_a^n + \gamma\bar{Q}_{\mathcal{M}}^{n-1} - \tilde{B}_C^n\right] \\
&\quad - \left(\mathbb{E}\bar{C}_{a^{*,n}} + \gamma\bar{Q}_{\mathcal{M}}^{n-1}\right) \\
&= \mathbb{E}^{n-1}\left[\max_{a\in\mathcal{A}} \bar{C}_a^n - \tilde{B}_C^n\right] - \mathbb{E}\bar{C}_{a^{*,n}}\,. \qquad (25)
\end{aligned}
$$

In SS-MDP, the bias correction term $\tilde{B}_C^n$ can be written as $\tilde{B}_C^n = (\frac{\xi}{b}+b)\sigma_{a^{*,n}}$ with $b := b_{\mathcal{A}}$ and $\sigma_{a^{*,n}} := \bar{\sigma}(s^n, a^n)$. With this, we rearrange the term $\max_{a\in\mathcal{A}} \bar{C}_a^n - \tilde{B}_C^n$ in (25), using the abbreviation $\mu_{a^{*,n}} := \mathbb{E}\bar{C}_{a^{*,n}}$, as follows:

$$
\begin{aligned}
& \max_{a\in\mathcal{A}} \bar{C}_a^n - \tilde{B}_C^n \\
&= \max_{a\in\mathcal{A}} \bar{C}_a^n - \left(\left(\frac{\xi}{b}+b\right)\sigma_{a^{*,n}}\right) \\
&= \max_{a\in\mathcal{A}} \left(\bar{C}_a^n - \mu_{a^{*,n}}\right) - \left(\frac{\xi}{b}+b\right)\sigma_{a^{*,n}} + \mu_{a^{*,n}} \\
&= \sigma_{a^{*,n}}\left(\max_{a\in\mathcal{A}}\left(\frac{\bar{C}_a^n - \mu_{a^{*,n}}}{\sigma_{a^{*,n}}}\right) - b - \frac{\xi}{b}\right) + \mu_{a^{*,n}} \\
&= \frac{\sigma_{a^{*,n}}}{b}\left(b\left(\max_{a\in\mathcal{A}}\left(\frac{\bar{C}_a^n - \mu_{a^{*,n}}}{\sigma_{a^{*,n}}}\right) - b\right) - \xi\right) + \mathbb{E}\bar{C}_{a^{*,n}}\,. \qquad (26)
\end{aligned}
$$

Substituting (26) back into (25) gives

$$
\begin{aligned}
& \mathbb{E}^{n-1}\left[\max_{a\in\mathcal{A}} \bar{C}_a^n - \tilde{B}_C^n\right] - \mathbb{E}\bar{C}_{a^{*,n}} \\
&= \mathbb{E}^{n-1}\left[\frac{\sigma_{a^{*,n}}}{b}\left(b\left(\max_{a\in\mathcal{A}}\left(\frac{\bar{C}_a^n - \mu_{a^{*,n}}}{\sigma_{a^{*,n}}}\right) - b\right) - \xi\right)\right. \\
&\qquad \left. + \mathbb{E}\bar{C}_{a^{*,n}}\right] - \mathbb{E}\bar{C}_{a^{*,n}} \\
&= \frac{\sigma_{a^{*,n}}}{b}\left(\mathbb{E}^{n-1}\left[b\left(\max_{a\in\mathcal{A}}\left(\frac{\bar{C}_a^n - \mu_{a^{*,n}}}{\sigma_{a^{*,n}}}\right) - b\right)\right] - \xi\right). \qquad (27)
\end{aligned}
$$

Let $Y_a := b(\max_{a\in\mathcal{A}}(\frac{\bar{C}_a^n - \mu_{a^{*,n}}}{\sigma_{a^{*,n}}}) - b)$. As $\bar{C}_a^n$ is a sample average of $C(a)$ and $|C(a)| < C^M < \infty$, we note that $\sup_a \mathbb{E}[|\max_{a\in\mathcal{A}} \bar{C}_a^n|^2] < \infty$. This implies uniform integrability of Y_a.

With uniform integrability of Y_a and Lemma 11, which shows that Y_a converges to standard Gumbel distribution as $|\mathcal{A}| \to \infty$, Y_a converges in mean to the mean of standard Gumbel random variable ξ. The RHS of (27) tends to 0 as $|\mathcal{A}| \to \infty$, and therefore $\tilde{B}_C^n \xrightarrow{|\mathcal{A}|\to\infty} \bar{B}^{*,n}$. As a note, assuming the normality of $\bar{C}_a^n$ is generally reasonable when n is large due to a central limit theorem on bounded finite variance random variable C. ■

This theorem requires $|\mathcal{A}| \to \infty$ for the bias correction term $\bar{B}^n$ to asymptotically converge to the correct bias B^n in mean. In practice, where a finite action space is given, $\bar{B}^n$ tend to be larger than, so the resulting $\hat{Q}_{BC}^n$ is overcorrected and it has smaller value than the optimally corrected Q-sample $\hat{Q}^{*,n}$. Theoretically, this overestimation of bias gets worse with

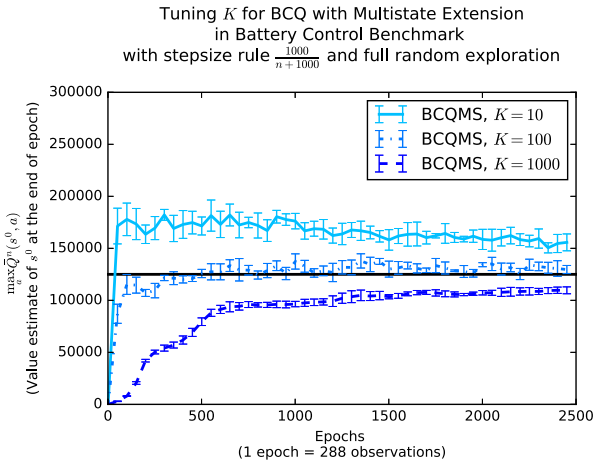

Fig. 2. This plot shows the effect of tuning the parameter K of BCQ-MS. Simulation settings are held the same as those used to generate Fig. 4. We report the mean and the standard deviation from five independent runs, and match the *y*-axis of the plot with that of the plot in Fig. 4. Larger K means stronger correction against max-operator bias due to random transition. The optimal value is plotted as a thick horizontal line around 1 20 000.

smaller action space; yet we believe that this issue has much smaller effect in practice. When the initial values are set to be lower than the optimal value, the oversized correction term may slow down the empirical convergence from below, but this slowdown is much milder than what happens when letting the max-operator bias propagate through max operators in Q-learning algorithm. We demonstrate the empirical results in Sections V and VI, where the empirical effect of undercorrection and overcorrection of max-operator bias on the Q-estimate is shown in Fig. 2 in Section VI.

IV. Multistate Extension of BCQ

Of the two different sources of max-operator bias in Q-learning as defined in (11), BCQ algorithm for SS-MDP shown in Algorithm 1 helps to correct only the B_C^n term. We lift this limitation, and construct a multistate extension to BCQ so that it can handle max-operator bias in MDPs with multiple states. The multistate extension adds another bias correction term that targets the B_T^n term of max-operator bias in Q-learning.

A. Designing the Correction Term for B_T^n

We aim to correct the bias due to stochastic transition, denoted as B_T^n and defined in (13). To do so, we take a direct approximation of the expectation in (13) with the empirical expectation, which leads to the multistate bias-correction term $\tilde{B}_T^n$ defined as

$$\tilde{B}_T^n := \gamma\left(\max_{a'\in\mathcal{A}(s^{n+1})}\left(\bar{Q}^{n-1}\left(s^{n+1},a'\right)\right) - \frac{1}{K}\sum_{k=1}^{K}\left(\max_{a'\in\mathcal{A}\left(\hat{\mathcal{T}}^k\right)}\bar{Q}^{n-1}\left(\hat{\mathcal{T}}^k,a'\right)\right)\right) \tag{28}$$

where $\hat{\mathcal{T}}^k$ are elements of a set of K most recent transition observations from s^n. That is, $\hat{\mathcal{T}}^k$ are the elements of a set comprising K last elements of set $\mathcal{T}_K := \{s^i|(s^{i-1},a^{i-1}) = (s^n,a^n)\}_{1\le i\le n+1}$ when the elements are sorted by increasing order of i. When the set $\mathcal{T}_K$ has size less than K, then the set is augmented to have size K with dummy state elements s_ϕ. The dummy state s_ϕ is an arbitrary state whose value estimate $\bar{Q}^{n-1}(s_\phi,a)$ is defined to be the initialization value $\bar{Q}^0$.

Algorithm 2: BCQ Algorithm With Multistate Extension.

Require: $\bar{Q}^0(s,a)$, s^0, γ, access to MDP $(\mathcal{S},\mathcal{A},C,T)$, N, stepsize rule α_n, Burn-in parameter K

1: **for** $n=0,1,\ldots,N-1$ **do**
2: Decide a^n
3: Observe $\hat{C}^{n+1}\sim C(s^n,a^n)$
4: Observe $s^{n+1}=\hat{T}^{n+1}\sim T\left(s^{n-1},a^{n-1}\right)$
5: **if** $\tilde{B}_C^{n+1}$ is computable and $\tilde{B}_T^{n+1}$ is computable **then**
6: Compute $\tilde{B}_C^{n+1}\leftarrow\left(\frac{\xi}{b_{|\mathcal{A}(s^n)|}}+b_{|\mathcal{A}(s^n)|}\right)\bar{\sigma}_n(s^n,a^n)$
7: Compute $\tilde{B}_T^{n+1}\leftarrow\gamma(\max_{a'\in\mathcal{A}(s^{n+1})}(\bar{Q}^n(s^{n+1},a')) - \frac{1}{K}\sum_{k=1}^{K}(\max_{a'\in\mathcal{A}(\hat{\mathcal{T}}^k)}\bar{Q}^n(\hat{\mathcal{T}}^k,a')))$
8: Compute $\hat{Q}_{BC}^{n+1}\leftarrow\hat{C}^{n+1}(s^n,a^n)+\gamma\max_{a\in\mathcal{A}(s^{n+1})}\bar{Q}^n\left(s^{n+1},a\right)-(\tilde{B}_C^n+\tilde{B}_T^n)$
9: Update $\bar{Q}^{n+1}(s^n,a^n)\leftarrow(1-\alpha_n(s^n,a^n))\bar{Q}^n(s^n,a^n)+\alpha_n(s^n,a^n)\hat{Q}_{BC}^{n+1}$
10: **end if**
11: **end for**
12: **return** $\bar{Q}^N(s,a)$

B. Structural Property of Multistate Bias Correction Term

To have bias correction against B_T^n, K can be set to any integer greater than one. The larger K, the greater the maximum stability of bias correction against stochastic transition, at the cost of greater space complexity to keep track of $\mathcal{T}_K$. As $K\to\infty$, the bias correction becomes asymptotically optimal since $\tilde{B}_T^n\to B_T^n$. Meanwhile, setting $K=1$ results in a degenerate case in which there will be no bias correction effect because for $K=1$, $\tilde{B}_T^n=0$.

C. BCQ Algorithm With Multistate Extension

The BCQ-MS is shown in Algorithm 2. It extends Algorithm 1 by applying the multistate bias-correction term. The key property of multistate extension is the bias correction term $\tilde{B}_T^n$, which is incorporated into the update step of $\bar{Q}^n$.

The condition for "$\tilde{B}_T^n$ is computable," similar to that for $\tilde{B}_C^n$, true if at least K samples of $\hat{T}^i$ where $\left(s^i,a^i\right)=\left(s^n,a^n\right)$ for $i\le n$ is available, where the new input parameter K determines the relative strength of multistate bias correction. The value of K can start as low as two, but a larger value may be necessary to provide sufficient bias-correction depending on the state space size and transition matrix of the underlying MDP. A general guideline for setting the parameter K is to set it as at least the number of states that can be reached in a single step from any state.

As a rule of thumb, BCQ-MS will benefit from larger stepsizes than the classic Q-learning algorithm, which often suffers from max-operator bias with large stepsizes. Therefore, for faster empirical convergence, we recommend the stepsize rule α_n for BCQ-MS to be larger than what is used in Q-learning.

D. Asymptotic Convergence of BCQ-MS

We show that the BCQ-MS produces the same Bellman optimal result as Q-learning. The proof of asymptotic convergence of the BCQ is comprised of three steps, corresponding to the three stages depending on the iteration counter n. First, we show that the Q values after the first (burn-in) stage have finite deviation from any given initial setting satisfying the set of requirements for vanilla Q-learning to converge, as given in [2]. Second, we show that the Q values after the second (bias-correction) stage have finite deviation. Third, we show that the snapshot of any configuration of all parameters of the BCQ algorithm after the second stage eventually converges to produce Bellman optimal output $\bar{Q}^n$ using a Martingale noise sequence. We present three lemmas in a row, each of which corresponds to the steps outlined above.

We provide two stage-delimiting parameters to be used in this section to guide the convergence proof.

1) N^B: "Begin" iteration threshold. When the iteration counter $0 \leq n < N^B$, the BCQ is in burn-in stage, where it is using samples to estimate variables necessary to compute the bias correction term via bootstrapping. Unless it is arbitrarily tuned, $N^B = n$ when the condition "$\tilde{B}_C^n$ is computable and $\tilde{B}_T^n$ is computable" is first satisfied in Algorithm 2.
2) N^E: "End" iteration threshold. When the iteration counter $N^B \leq n \leq N^E$, the BCQ is in its bias-corrected learning phase. Bias correction takes place to reduce max-operator induced bias in learning trajectory of Q estimate. Unless arbitrarily set, $N^E = N - 1 < \infty$ is assumed for implementation of Algorithm 2. However, in implementations where the iteration counter $n > N^E$, we assume that bias correction is no longer applied.

The above-mentioned parameters can be set by the user to fit the nature of intended convergence pattern for the BCQ. In our implementation used to perform experiments reported in later sections, we use $N^B = 2$, the first threshold for burn-in stage. Also in our implementation, we set $N^E = N - 1$, since we demonstrate the effectiveness of bias correction and the evolution of value estimates over the entire run of N observations.

The following is the set of assumptions required for the asymptotic convergence of BCQ to Bellman optimal condition.

(A1) The policy to choose a^n given $\bar{Q}^{n-1}$ and s^n is well behaved, such that all state-action pairs in $\mathcal{S} \times \mathcal{A}$ are visited infinitely often.

(A2) The stepsize rule α conforms to the following conditions:

$$\sum_n \alpha_{n-1}\left(s^n, a^n\right) = \infty \qquad \text{with probability 1}$$

$$\sum_n \left(\alpha_{n-1}\left(s^n, a^n\right)\right)^2 < \infty \qquad \text{with probability 1}$$

for any sample realization of (s^n, a^n).

(A3) The sampled reward $\bar{C}^n(s,a)$ follows the distribution of $C(s,a)$ satisfying the following conditions:

$$\mathbb{E}C(s,a) < \infty$$

$$\mathbb{E}\left(C(s,a)\right)^2 < \infty$$

for $n = 0, 1, \ldots$ and any sample realization of (s,a).

(A4) An arbitrary initial value of $\bar{Q}^0(s,a)$ is finite for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

(A5) The discount factor γ satisfies $0 \leq \gamma < 1$.

This set of assumptions (A1–A5) is sufficient to show the asymptotic convergence of BCQ to the same Bellman optimal condition as Q-learning.

Lemma 13 (On $\hat{Q}^n$): $\forall n = 1, 2, \ldots$, $\hat{Q}^n < \infty$ with probability 1, given $\bar{Q}^{n-1} < \infty, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$, where $\hat{Q}^n$ is defined as

$$\hat{Q}^n := \bar{C}^n\left(s^n, a^n\right) + \gamma \max_{b \in \mathcal{A}(s^{n+1})} \bar{Q}^{n-1}\left(s^{n+1}, b\right). \tag{29}$$

Proof: We provide a proof by induction. Begin by assuming that for $n = 1$, we have

$$\hat{Q}^1 := \bar{C}^1 + \gamma \max_{b \in \mathcal{A}(s^1)} \bar{Q}^0\left(s^2, b\right)$$

where $\forall (s,a) \in \mathcal{S} \times \mathcal{A}, \bar{Q}^0 < \infty$ by Assumption A4. The term $\bar{C}^1 < \infty$ with probability 1 because $\hat{C}^1 \sim C\left(s^1, a^1\right)$ where $\forall (s,a) \in \mathcal{S} \times \mathcal{A}, \mathbb{E}C(s,a) < \infty$ by assumption A3. $\gamma < \infty$ by Assumption A5. Therefore, $\hat{Q}^1 < \infty$.

The inductive case also holds in a similar proof as follows:

$$\hat{Q}^n := \bar{C}^n + \gamma \max_{b \in \mathcal{A}(s^{n+1})} \bar{Q}^{n-1}\left(s^{n+1}, b\right)$$

where $\forall (s,a) \in \mathcal{S} \times \mathcal{A}, \bar{Q}^{n-1} < \infty$ by the inductive hypothesis, and the other terms $\bar{C}^{n-1}$ and γ are finite in the same manner as the base case proof. Therefore, $\hat{Q}^n < \infty$. ■

Lemma 14 (On Q estimate): $\forall n = 1, 2, \ldots$, $\bar{Q}^n < \infty$ with probability 1, given $\bar{Q}^{n-1} < \infty, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$, where $\bar{Q}^n := \left(1 - \alpha_{n-1}\left(s^n, a^n\right)\right) \bar{Q}^{n-1}\left(s^n, a^n\right) + \alpha_{n-1}\left(s^n, a^n\right) \hat{Q}^n$, $\hat{Q}^n$ is as used in Lemma 13, and α_{n-1} is the stepsize rule satisfying Assumption A2.

Proof: $\forall n = 1, 2, \ldots$, $\bar{Q}^{n-1} < \infty$ by assumption, and according to Lemma 13, $\hat{Q}^n < \infty$. The definitions of $\tilde{B}_C^n$ and $\tilde{B}_T^n$ implies the quantities to be finite, as the initial value $\bar{Q}^0$ is finite by Assumption A4. The definition of $\bar{Q}^n$ as shown above is a linear combination of $\bar{Q}^{n-1}$ and $\hat{Q}^n$, both of which are finite, so $\bar{Q}^n < \infty$. ■

Lemma 15 (Finite Divergence Phase 1): For any N_1, N_2 such that $0 \leq N_1 < N_2 < \infty$ and $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

$$\left|\bar{Q}^{N_2}(s,a) - \bar{Q}^{N_1}(s,a)\right| < \infty \quad \text{with probability 1}$$

given the assumptions A1–A5 for BCQ.

Proof: Without loss of generality, assume (s,a) as any arbitrary element in $\mathcal{S} \times \mathcal{A}$

$$\begin{aligned}\left|\bar{Q}^{N_2}(s,a) - \bar{Q}^{N_1}(s,a)\right| &\leq \left|\bar{Q}^{N_2}(s,a)\right| + \left|\bar{Q}^{N_1}(s,a)\right| \\ &\leq |C_{N_2}| + |C_{N_1}| \\ &< \infty\end{aligned}$$

with probability 1. The last line mentioned above uses the result of Lemma 14, $\forall n = 1, 2, \ldots, \bar{Q}^n < \infty$, to show that the following arbitrary finite numbers C_{N_2} and C_{N_1} exist:

$$\exists C_{N_2} \text{ such that } \bar{Q}^{N_2}(s,a) < C_{N_2} < \infty, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$$

$$\exists C_{N_1} \text{ such that } \bar{Q}^{N_1}(s,a) < C_{N_1} < \infty, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$$

with probability 1. ■

Lemma 16 (On Q Estimate With Bias Correction): $\forall n = 2, 3, \ldots$, $\bar{Q}^n_{\text{BCQ}} < \infty$ with probablity 1, given $\bar{Q}^{n-1}_{\text{BCQ}} < \infty, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$, where $\bar{Q}^n_{\text{BCQ}}$ is defined as

$$\bar{Q}^n_{\text{BCQ}} := (1 - \alpha_{n-1}(s^n, a^n))\, \bar{Q}^{n-1}_{\text{BCQ}}(s^n, a^n) + \alpha_{n-1}(s^n, a^n) \left(\hat{Q}^n - \left(\tilde{B}^n_C + \tilde{B}^n_T \right) \right). \quad (30)$$

$\hat{Q}^n$ is defined as in (29), and α_{n-1} is the stepsize rule satisfying Assumption A2.

Proof: $\forall n = 2, 3, \ldots$, $\tilde{B}^n_C(s,a) < \infty$, since $\text{Var}\bar{C}(s,a) < \infty$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. Also, $\bar{Q}^{n-1}_{\text{BCQ}} < \infty$ by assumption, and according to lemma 13, $\hat{Q}^n < \infty$. The definition of $\bar{Q}^n_{\text{BCQ}}$ as shown above is a linear combination of the two finite values $\bar{Q}^{n-1}$ and $\hat{Q}^n$, so $\bar{Q}^n_{\text{BCQ}} < \infty$. ■

Lemma 17 (Finite Divergence Phase 2): For any N_2, N_3 such that $1 < N_2 < N_3 < \infty$ and $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

$$\left| \bar{Q}^{N_3}_{\text{BCQ}}(s,a) - \bar{Q}^{N_2}_{\text{BCQ}}(s,a) \right| < \infty \quad \text{with probability 1}$$

given Assumptions A1–A5 for BCQ, and $\bar{Q}^n_{\text{BCQ}}$ as defined in (30) for $n > 1$. ■

Proof: Apply Lemma 15 with $N_1 \leftarrow N_2$ and $N_2 \leftarrow N_3$.

Theorem 18 (Asymptotic Convergence of BCQ): For any arbitrary parameter settings such that $0 < N^B < N^E < \infty$ and satisfying Assumptions A1–A5, the output $\bar{Q}^n$ of BCQ from iteration n has the following asymptotic property:

$$\bar{Q}^n_{\text{BCQ}} \xrightarrow{n \to \infty} \bar{Q}^*$$

where $\bar{Q}^*$ is the same Bellman optimal convergence point for both Q-learning and BCQ.

Proof: We show that the divergence of $\bar{Q}^{N^E}$ from $\bar{Q}^0$ is at most finite for any parameter settings for BCQ under the set of assumptions A1–A5.

$$\begin{aligned} &\left| \bar{Q}^{N^E} - \bar{Q}^0 \right| \\ &= \left| \bar{Q}^{N^E} - \bar{Q}^{N^B - 1} + \bar{Q}^{N^B - 1} - \bar{Q}^0 \right| \\ &\leq \left| \bar{Q}^{N^E} - \bar{Q}^{N^B - 1} \right| + \left| \bar{Q}^{N^B - 1} - \bar{Q}^0 \right| \\ &= \left| \bar{Q}^{N^E}_{\text{BCQ}} - \bar{Q}^{N^B - 1}_{\text{BCQ}} \right| + \left| \bar{Q}^{N^B - 1} - \bar{Q}^0 \right| < \infty \end{aligned}$$

where the last equality is due to the fact that only from iteration N^B, bias-corrected update rule is applied (note that $\bar{Q}^{N^B - 1}_{\text{BCQ}} = \bar{Q}^{N^B - 1}$ in this case). Then, the last inequality comes out from applying Lemmas 17 and 15 for the first and the second deviation terms, respectively.

Since there is at most finite divergence of $\bar{Q}^{N^E}$ from the initial value $\bar{Q}^0$, we can always construct a new Q-learning instance with initial value set as $\bar{Q} = \bar{Q}^{N^E}$ and the same set of assumptions (A1–A5). This new instance of Q-learning is known to asymptotically converge to Bellman optimality under Assumptions A1–A5 in [2]. ■

Theorem 18 shows that BCQ asymptotically converges to the same point as Q-learning. This serves as a sanity check for the new algorithm such that it retains the key property of Q-learning, while it curtails the max-operator bias in Q-learning. This is not a proof of faster asymptotic rate of convergence, as the bias correction is active during a finite number of earlier iterations.

V. Multiarmed Bandit Simulation Benchmark

In this section, we use the game of Roulette as a benchmark, since the optimal policy (and value) is known. Roulette is a well-known gambling problem, in which players place different kinds of bets on a set of numbers and then receive a set contribution if the random outcome of Roulette spin matches the bet. At the same time, Roulette is an SS-MDP, where its stochastic contribution function gives rise to significant max-operator bias when classic Q-learning is applied. We use Roulette to show that the BCQ algorithm is highly resistant to the Q value overestimation found in the classic Q-learning algorithm.

A. Roulette as a Multiarmed Bandit Problem

Roulette can be naturally cast as a multiarmed bandit problem with 153 arms. In particular, 153 arms stand for the following possible types of bets: 38 actions of betting on 1 number, 29 actions on 2 numbers, 12 actions on 3 numbers, 22 actions on 4 numbers, 1 action on 5 ("top line") numbers, 6 actions on 6 numbers, 6 actions on 12 numbers, 6 on 18 numbers, and 1 on betting on nothing. These actions are grouped in their winning probabilities, and the order given above is in decreasing variance of the contribution function. These contribution functions are stochastic with varying degrees of noise inherent to different types of betting actions because each type of betting action has different winning odds. In an ideal Roulette, the contribution function is fully known, so the best cost-to-go function and the best policy can be computed *a priori*.

We simplify the game of Roulette as follows. We limit the betting strategy of Roulette by allowing the bets to be only \$0 or \$1, and keeping only 39 actions, which corresponds to 38 actions of betting on each of 38 numbers in Roulette and 1 no-betting action. In this case, 38 actions of betting have expected reward of $-\$0.0526$ and the no-betting action of \$0. We cast this simplified version of Roulette into a 39-armed bandit problem, whose optimal policy is not to play, or choosing the "no-betting" action. The simplified Roulette can be also seen as an SS-MDP, as after each roll of Roulette the agent must decide the action in turn.

B. Performance Comparison

When the simplified Roulette is seen as an infinite horizon Markov decision problem, the optimal policy is to select the no-betting action, and the optimal value function has value of 0, regardless of the value of the discount factor γ. We apply

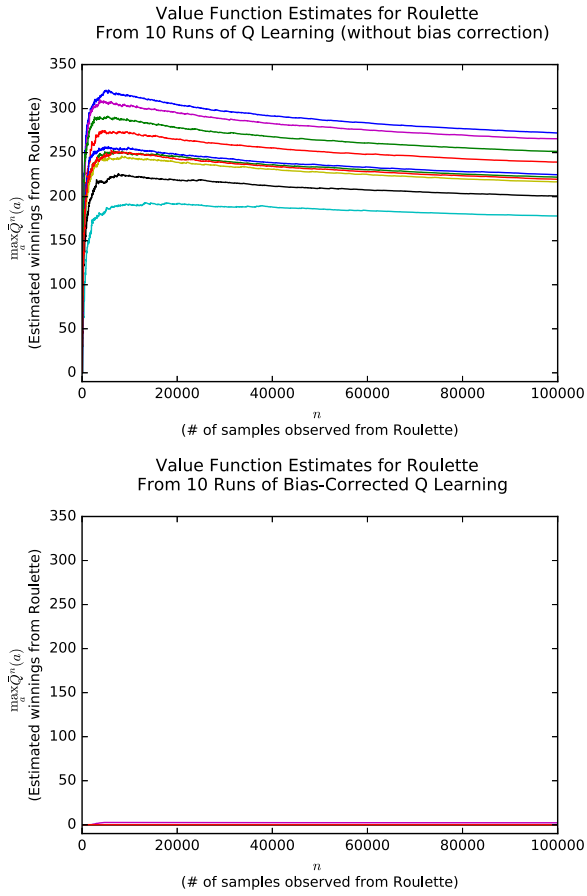

Fig. 3. Plot shows the evolution of value estimate from classic Q-learning and BCQ for SS-MDP, total of ten sample runs. Each figure has ten lines corresponding each sample run, which signifies consistently reproduced upward bias in Q learning in the top figure, and the suppression of it in the bottom figure. Note that 8 runs from the bottom figure, where bias correction is in action, are on top of each other as they are exactly on 0.

Q-learning and BCQ to this simplified Roulette, with $\gamma = 0.99$. A harmonic stepsize function $\alpha_{n-1} = \frac{100}{n+100}$ and pure random exploration policy to decide a^{n-1} are used in both algorithms. The initial value of $\bar{Q}^0$ is set to 0, which is the optimal value $\bar{Q}^*$. The default value of two for burn-in constant of BCQ is used.

We report the evolution of the value function estimate from Q^n estimates in Fig. 3. Q-learning shows massive overestimation of values for every ten runs, implying that it believes in winning a few hundred dollars in this benchmark Roulette that is modeled as $\gamma = 0.99$ discounted infinite horizon MDP. On the other hand, the BCQ estimate stays at 0 for eight of the ten repeats, and shows well-controlled (more than $1000\times$ reduction) value overestimation for the other two cases, where bias observed was not perfectly controlled due to approximate nature of bias correction.

VI. Electricity Storage Problem

There is growing interest in using grid-scale batteries to store energy to take advantage of electricity price variations and to smooth out generation from renewables. The problem can be formulated as finding a policy that maximizes the γ-discounted sum of rewards over time. We formulate the problem as a dynamic program with a state variable that captures the price of electricity and the amount of energy in storage. The stochastic volatility of electricity prices is the primary cause of the max-operator bias. We show that the max-operator bias is significant in Q-learning, but can be controlled using the multistate extension of BCQ.

A. Problem Formulation as an MDP

We model the electricity storage problem as an MDP, whose specification we discuss in this section. The state at time n is the minimally sufficient information to simulate the state at time $n+1$ in an MDP with a given policy π. Since we use the tabular representation of Q values, the state at time n, denoted s^n, is comprised of the following discrete values: The binned current storage level $b^R(R^n)$, and the binned current electricity spot price level $b^P(p^n)$. The binning function b^R discretizes the storage levels as 11 evenly spaced linear space with max value of 2000 and min value of 0, and the binning function b^P discretizes the price levels as 11 evenly-spaced log-space with max and min value set as the maximum and minimum observed prices.

The action at time n, denoted a^n, is the amount of energy traded. The actual values of a^n is discretized by choosing one from a set of possible values that exactly fits the difference of two bin averages of energy stored in the battery. Therefore, we note that the following holds for all n:

$$a^n := b^R\left(R^{n+1}\right) - b^R\left(R^n\right). \tag{31}$$

The contribution function is set to be the actual cost to buy and revenue to sell electricity to the spot market at the current market price of electricity. Formally stated

$$\hat{C}\left(s^n, a^n\right) := -p^n a^n \tag{32}$$

$$= -p^n\left(b^R\left(R^{n+1}\right) - b^R\left(R^n\right)\right) \tag{33}$$

where p^n is the undiscretized spot price of a unit of electricity at time n. It is important to note that the discretization in the state variable creates additional stochasticity in the cost function stated above, as the binned representation of electricity price in state variable will create the following effect: Given a state s' and a suitable action a', there exist two distinct price values $p_1 \neq p_2$ such that $b^P(p_1) = b^P(p_2)$. This implies $\mathrm{Var}[\hat{C}(s^n, a^n)|s', a'] > 0$, which in turn implies $\mathrm{Var}^{n-1}[\hat{C}^n] > 0$ in the Q-learning and the BCQ algorithms applied to solve the MDP.

To model the transition of electricity price p^n as n increases, we fit a hidden semi-Markov model to a real-world data collected from grids near Princeton, NJ, USA, which is shown to predict the crossing time distributions of the actual price processes much better than autoregression models, such as an ARMA model [20].

The discount factor of $\gamma = 0.95$ is used, and the MDP is modeled in 5-min increments over a 1-d horizon, which translates to 288 time periods per epoch.

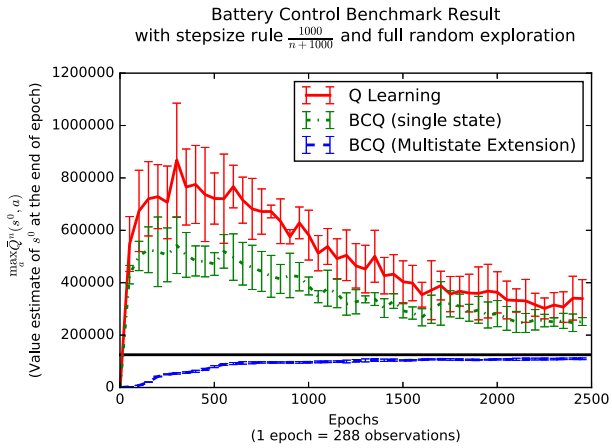

Fig. 4. This plot shows the evolution of value function estimate from three different algorithms. Five independent runs are averaged and plotted, with the standard deviation values plotted as error bars. The optimal value is plotted as a thick horizontal line around 1 20 000.

B. Performance Comparison

We first demonstrate the effectiveness of multistate extension in controlling max-operator bias in this battery benchmark problem. We present the evolution of the value estimate of the initial state of epochs, from epoch 0 to epoch 2499, for the following algorithms.

1) Q-learning (no correction against either stochastic contribution or state transition).
2) BCQ for SS-MDP (no correction against state transition).
3) BCQ-MS, strong correction $K = 1000$.

For all cases, we use the stepsize rule of $\alpha_n = \frac{1000}{n+1000}$, initial value $\bar{Q}^0 = 0$, and pure exploration policy to cover the state-action space as evenly as possible. We repeat five independent runs of the algorithms, and report the mean value estimate and its standard deviation in Fig. 4.

Clearly, BCQ-MS shows effective resistance against the max-operator bias in battery control benchmark problem. On the other hand, Q-learning overshoots value estimation due to the max-operator bias. BCQ for SS-MDP, which by design, cannot effectively counteract bias due to transition in multistate MDPs, shows similar overshoot as Q-learning.

We also present the effect of varying bias-correction strength in BCQ-MS. To do so, we use $K \in \{10, 100, 1000\}$ in the battery benchmark problem and report the evolution of value estimates from the following algorithms in Fig. 2. Using smaller K reduces bias-correcting effect of BCQ-MS and the value estimates may overshoot as shown in $K = 10$ case, but that is nowhere as significant as Q-learning and BCQ for SS-MDP shown in Fig. 4. For the ease of comparison between Figs. 2 and 4, we maintain other factors the same: We use the stepsize rule of $\alpha_n = \frac{1000}{n+1000}$, initial value $\bar{Q}^0 = 0$, and pure exploration policy to cover the state-action space as evenly as possible.

VII. Discussion

BCQ for SS-MDP rely on $|\mathcal{A}| \to \infty$ for its optimal correction property, and it overcorrects the bias in SS-MDPs with finite action space. Also, in its multistate extension, the tunable parameter K also introduces the potential for overcorrection when K values are set too high. We consider that overcorrection is possible, but it has much smaller impact on the convergence of Q-estimate than the letting the max-operator bias to propagate through with undercorrection. As shown in Fig. 2, the overcorrected $K = 1000$ trajectory closes the gap from the optimal much faster than the undercorrected $K = 10$ trajectory, which has a mild overshoots. This is because the max operator in Q-learning filters out underestimated Q values, whereas the overestimated Q values tend to propagate to Q-values of other state-action pairs during learning and delays the convergence once the overshooting happens. Therefore, we think that the impact of overcorrecting bias due to violating $|\mathcal{A}| \to \infty$ assumption of bias-correction term is mild in practice because the overcorrection takes much less number of samples to recover than undercorrection.

We note that the bias-correction machinery in BCQ-MS introduces additional computational burden compared to Q-learning. In theory, per-iteration time complexity is increased by $O(1)$ due to computing the correction terms, and the space complexity is increased by $O(|\mathcal{S}||\mathcal{A}|K)$ due to computing the estimators for C and σ, as well as computing multistate extension part. In particular, in the experiments we performed, extra computational burden of bias correction increased per-iteration runtime approximately by 40% compared to Q-learning, when executed to reach 1 million iterations. Considering that Q-learning may take much more iterations to correct max-operator bias once it shows up in its $\bar{Q}$ estimate as shown in Fig. 4, it makes a sensible decision to avoid that path at the cost of increasing per-iteration runtime by 40%.

We presented asymptotic convergence result of BCQ-MS, as we consider this being the fundamental property for a correction to address the inherent problem in Q-learning algorithm. A natural direction for future research that deserves theoretical analysis is the finite time rate analysis on the error in $\bar{Q}^n_{\text{BCQ}}$ in comparison to $\bar{Q}^n$, especially once it gains positive bias due to max-operator bias in $\hat{Q}^n$.

VIII. Conclusion

We identified the sample bias caused by max operator in Q-learning algorithm as the max-operator bias, and characterized the bias into two additive terms by their origin of stochasticity. We presented correction methods for each of the two bias terms, with which we design a plug-in addition to augment Q-learning algorithm to have resistance against max-operator bias, resulting in BCQ-MS. We demonstrated that Roulette and Electricity Storage MDPs are useful benchmark problems where max-operator bias is easily observed in Q-learning. We showed that each bias correction method introduced has intended effects of controlling two sources of max-operator bias, as demonstrated in the value estimates from the two benchmark problems.

References

[1] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Dept. Psychol., King's College, London, U.K., 1989.
[2] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," *Mach. Learn.*, vol. 16, pp. 185–202, 1994, doi: 10.1007/BF00993306. [Online]. Available: http://dx.doi.org/10.1007/BF00993306

[3] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Comput.*, vol. 6, no. 6, pp. 1185–1201, 1994.
[4] V. S. Borkar and S. P. Meyn, "The O.D.E. method for convergence of stochastic approximation and reinforcement learning," *SIAM J. Control Optim.*, vol. 38, no. 2, pp. 447–469, Jan. 2000. [Online]. Available: http://dx.doi.org/10.1137/S0363012997331639
[5] C. Szepesvari, "The asymptotic convergence rate of Q-learning," in *Proc. Neural Inf. Process. Syst.*, vol. 10, 1997, pp. 1064–1070. [Online]. Available: www.ualberta.ca/∼szepesva/papers/nips97.ps.pdf
[6] M. Kearns and S. Singh, "Finite-sample convergence rates for Q-learning and indirect algorithms," in *Proc. Neural Inf. Process. Syst.*, vol. 12, 1999, pp. 996–1002.
[7] E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," *J. Mach. Learn. Res.*, vol. 5, pp. 1–25, 2003. [Online]. Available: http://jmlr.csail.mit.edu/papers/volume5/evendar03a/evendar03a.pdf
[8] K.-H. Park, Y.-J. Kim, and J.-H. Kim, "Modular Q-learning based multi-agent cooperation for robot soccer," *Robot. Auton. Syst.*, vol. 35, no. 2, pp. 109–122, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889001001142
[9] J. Yao, J. Chen, and Z. Sun, "An application in robocup combining Q-learning with adversarial planning," in *Proc. 4th World Congr. Intell. Control Autom.*, vol. 1, 2002, pp. 496–500.
[10] K. Scheffler and S. Young, "Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning," in *Proc. 2nd Int. Conf. Human Lang. Technol. Res.*, 2002, pp. 12–19. [Online]. Available: http://dl.acm.org/citation.cfm?id=1289189.1289246
[11] G. Tesauro and J. O. Kephart, "Pricing in agent economies using multi-agent Q-Learning," *Auton. Agents Multi-Agent Syst.*, vol. 5, pp. 289–304, 2002, doi: 10.1023/A:1015504423309. [Online]. Available: http://dx.doi.org/10.1023/A:1015504423309
[12] G.-S. Yang, E.-K. Chen, and C.-W. An, "Mobile robot navigation using neural Q-learning," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 1, Aug. 2004, pp. 48–52.
[13] T. Graepel, R. Herbrich, and J. Gold, "Learning to fight," in *Proc. Int. Conf. Comput. Games: Artificial Intell., Des. Educ.*, 2004, pp. 193–200. [Online]. Available: http://research.microsoft.com/pubs/65639/graehergol04.pdf
[14] Y.-C. Wang and J. M. Usher, "Application of reinforcement learning for agent-based production scheduling," *Eng. Appl. Artif. Intell.*, vol. 18, no. 1, pp. 73–82, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197604001034
[15] D. V. Djonin, "Q-learning algorithms for constrained Markov decision processes with randomized monotone policies: Application to mimo transmission control," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2170–2181, May 2007. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4156378
[16] D. Ormoneit and Ś. Sen, "Kernel-based reinforcement learning," *Mach. Learn.*, vol. 49, no. 2, pp. 161–178, Nov. 2002. [Online]. Available: https://doi.org/10.1023/A:1017928328829
[17] H. P. van Hasselt, "Double q-learning," *Advances Neural Inf. Process. Syst.*, vol. 23, pp. 2613–2621, 2010. [Online]. Available: http://books.nips.cc/papers/files/nips23/NIPS2010_0208.pdf
[18] D. Lee, B. Defourny, and W. B. Powell, "Bias-corrected Q-learning to control max-operator bias in Q-learning," in *Proc. IEEE Symp. Adaptive Dyn. Program. Reinforcement Learn.*, Apr. 2013, pp. 93–99.
[19] D. Lee and W. B. Powell, "An intelligent battery controller using bias-corrected Q-learning," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 316–322. [Online]. Available: http://dl.acm.org/citation.cfm?id=2900728.2900774
[20] J. Durante, R. Patel, and W. B. Powell, "Time series methods for replicating crossing times in spatially distributed stochastic systems," to be published.

Donghun Lee received the B.A. degree in biochemistry from Columbia University, New York, NY, USA, in 2007, and an M.S. degree in computational biology from Carnegie Mellon University, Pittsburgh, PA, USA, in 2009. He is working toward the Ph.D. degree in computer science in the Department of Computer Science at Princeton University, Princeton, NJ, USA, under the advisement of Professor Warren B. Powell.

He worked with Samsung Electronics from 2012 to 2016. His research interests include designing efficient learning algorithms in hierarchical online decision making problems.

Warren B. Powell (M'06) is a Professor in the Department of Operations Research and Financial Engineering at Princeton University, Princeton, NJ, USA, where he been teaching since 1981. He founded and directs CASTLE Labs (www.castlelab.princeton.edu), specializing in fundamental contributions to computational stochastic optimization with a wide range of applications. He has authored/coauthored over 200 publications and two books. His research interest includes computational stochastic optimization, with applications in energy, transportation, health, and finance.