Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Nonasymptotic Analysis of Monte Carlo Tree Search

Devavrat Shah, Qiaomin Xie, Zhi Xu

To cite this article:
Devavrat Shah, Qiaomin Xie, Zhi Xu (2022) Nonasymptotic Analysis of Monte Carlo Tree Search. Operations Research

Published online in Articles in Advance 01 Mar 2022

. https://doi.org/10.1287/opre.2021.2239

http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–27
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

Nonasymptotic Analysis of Monte Carlo Tree Search

Devavrat Shah,[a] Qiaomin Xie,[b,*] Zhi Xu[a]

[a] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139; [b] Industrial and Systems and Engineering, University of Wisconsin-Madison, Madison, Wisconsin 53706
*Corresponding author

Contact: devavrat@mit.edu, https://orcid.org/0000-0003-0737-3259 (DS); qiaomin.xie@wisc.edu, https://orcid.org/0000-0003-2834-6866 (QX); zhixu@mit.edu, https://orcid.org/0000-0002-1421-2309 (ZX)

Received: January 6, 2020
Revised: April 16, 2021
Accepted: October 15, 2021
Published Online in Articles in Advance: March 1, 2022

Area of Review: Machine Learning and Data Science

https://doi.org/10.1287/opre.2021.2239

Abstract. In this work, we consider the popular tree-based search strategy within the framework of reinforcement learning, the Monte Carlo tree search (MCTS), in the context of the infinite-horizon discounted cost Markov decision process (MDP). Although MCTS is believed to provide an approximate value function for a given state with enough simulations, the claimed proof of this property is incomplete. This is because the variant of MCTS, the upper confidence bound for trees (UCT), analyzed in prior works, uses "logarithmic" bonus term for balancing exploration and exploitation within the tree-based search, following the insights from stochastic multiarm bandit (MAB) literature. In effect, such an approach assumes that the regret of the underlying recursively dependent nonstationary MABs concentrates around their mean exponentially in the number of steps, which is unlikely to hold, even for stationary MABs. As the key contribution of this work, we establish polynomial concentration property of regret for a class of *nonstationary* MABs. This in turn establishes that the MCTS with appropriate *polynomial* rather than *logarithmic* bonus term in UCB has a claimed property. Interestingly enough, empirically successful approaches use a similar polynomial form of MCTS as suggested by our result. Using this as a building block, we argue that MCTS, combined with nearest neighbor supervised learning, acts as a "policy improvement" operator; that is, it iteratively improves value function approximation for *all* states because of combining with supervised learning, despite evaluating at only finitely many states. In effect, we establish that to learn an ε approximation of the value function with respect to ℓ_∞ norm, MCTS combined with nearest neighbor requires a sample size scaling as $\tilde{O}(\varepsilon^{-(d+4)})$, where d is the dimension of the state space. This is nearly optimal because of a minimax lower bound of $\tilde{\Omega}(\varepsilon^{-(d+2)})$, suggesting the strength of the variant of MCTS we propose here and our resulting analysis.

Funding: This work was supported by the National Science Foundation [Grant CNS-1955997 and TRIPODS Phase II Grant] and MIT-IBM project on "Representation Learning as a Tool for causal Discovery," Siemens Futuremakers Fellowship.

Keywords: Monte Carlo tree search • Nonstationary multi-armed bandit • reinforcement learning

1. Introduction

Monte Carlo Tree Search (MCTS) is a search framework for finding optimal decisions based on the search tree built by random sampling of the decision space (Browne et al. 2012). MCTS has been widely used in sequential decision makings that have a tree representation, exemplified by games and planning problems. Since MCTS was first introduced, many variations and enhancements have been proposed. Recently, MCTS has been combined with deep neural networks for reinforcement learning, achieving remarkable success for games of Go (Silver et al. 2016, 2017b), chess, and shogi (Silver et al. 2017a). In particular, AlphaGo Zero (AGZ) (Silver et al. 2017b) uses supervised learning to learn a policy/value function (represented by a neural network) based on samples generated via MCTS; the neural network is recursively used to estimate the value of leaf nodes in the next iteration of MCTS for simulation guidance.

Despite the wide application and empirical success of MCTS, there is only limited work on theoretical guarantees of MCTS and its variants. One exception is the work of Kocsis and Szepesvári (2006) and Kocsis et al. (2006), which propose running a tree search by applying the upper confidence bound algorithm—originally designed for stochastic multiarm bandit (MAB) problems (Agrawal 1995, Auer et al. 2002)—to each node of the tree. This leads to the so-called upper confidence bounds for trees (UCT) algorithm, which is one of the popular forms of MCTS. In

Kocsis and Szepesvári (2006), a certain asymptotic optimality property of UCT is claimed. The proof therein is, however, incomplete, as we discuss in greater detail in Section 1.2. More importantly, UCT as suggested in Kocsis and Szepesvári (2006) requires exponential concentration of regret for the underlying nonstationary MAB, which is unlikely to hold in general even for stationary MAB as pointed out in Audibert et al. (2009).

Indeed, rigorous analysis of MCTS is subtle, although its asymptotic convergence may seem natural. A key challenge is that the tree policy (e.g., UCT) for selecting actions typically needs to balance exploration and exploitation, so the random sampling process at each node is nonstationary (nonuniform) across multiple simulations. A more severe difficulty arises because of the hierarchical/iterative structure of tree search, which induces complicated probabilistic dependency between a node and the nodes within its subtree. Specifically, as part of simulation within MCTS, at each intermediate node (or state), the action is chosen based on the outcomes of the past simulation steps within the subtree of the node in consideration. Such strong dependencies across time (i.e., depending on the history) and space (i.e., depending on the subtrees downstream) among nodes makes the analysis nontrivial.

The goal of this paper is to provide a rigorous theoretical foundation for MCTS. In particular, we are interested in the following:

- What is the appropriate form of MCTS for which the asymptotic convergence property claimed in the literature (Kocsis and Szepesvári 2006, Kocsis et al. 2006) holds?
- Can we rigorously establish the "strong policy improvement" property of MCTS when combined with supervised learning as observed in the literature (Silver et al. 2017b)? If yes, what is the quantitative form of it?
- Does supervised learning combined with MCTS lead to the optimal policy, asymptotically? If so, what is its finite-sample (nonasymptotic) performance?

1.1. Our Contributions

As the main contribution of this work, we provide affirmative answers to all of the previous questions. In what follows, we provide a brief overview of our contributions and results.

1.1.1. Nonstationary MAB and Recursive Polynomial Concentration. In stochastic MAB, the goal is to discover, among finitely many actions (or arms), the one with the best average reward while choosing as few nonoptimal actions as possible in the process. The rewards for any given arm are assumed to be independent and identically distributed (i.i.d.). The usual exponential concentration for such i.i.d. and hence stationary processes leads to the UCB algorithm with a *logarithmic* bonus term: at each time, choose an action with maximal index (ties broken arbitrarily), where the index of an arm is defined as the empirical mean reward plus constant times $\sqrt{\log t/s}$, where t is the total number of trials thus far, and $s \le t$ is the number of times the particular action is chosen in these t trials.

The goal in the MCTS is very similar to the MAB setup described previously: choose an action at a given query state that gives the best average reward. However, the reward depends on future actions. Therefore, to determine the best action for the given state, one has to take future actions into account, and MCTS does this by simulating future via effectively expanding all possible future actions recursively in the form of (decision-like) trees. In essence, the optimal action at the root of such a tree is determined by finding optimal path in the tree. Determining this optimal path requires solving multiple MABs, one per each intermediate node within the tree. Apart from the MABs associated with the lowest layer of the tree, all the MABs associated with the intermediate nodes turn out to have rewards that are the rewards generated by MAB algorithms for nodes downstream. This creates complicated, hierarchically interdependent MABs.

To determine the appropriate, UCB-like index algorithm for each node of the MCTS tree, it is essential to understand the concentration property of the rewards, that is, concentration of regret for MABs associated with nodes downstream. Although the rewards at leaf level may enjoy exponential concentration, because of independence, the regret of any algorithm for such an MAB is unlikely to have exponential concentration in general (Audibert et al. 2009, Salomon and Audibert 2011). Furthermore, the MAB of our interest has nonstationary rewards because of strong dependence across the hierarchy. Indeed, an oversight of this complication led Kocsis and Szepesvári (2006) and Kocsis et al. (2006) to suggest the UCT inspired by the standard UCB algorithm for MABs with stationary, independent rewards.

As an important contribution of this work, we formulate an appropriate form of nonstationary MAB that correctly models the MAB at each of the node in the MCTS tree. For such a nonstationary MAB, we define the UCB algorithm with an appropriate index and under which we establish appropriate concentration of the induced regret. This, in turn, allows us to recursively define the UCT algorithm for MCTS by appropriately defining index for each of the node-action within the MCTS tree. Here we provide a brief summary.

Given $[K] = \{1, \dots, K\}$ actions or arms, let $X_{i,t}$ denote the reward generated by playing arm $i \in [K]$ for the tth time. Let empirical mean over n trials for arm i be $\bar{X}_{i,n} = \frac{1}{n}\sum_{t=1}^{n} X_{i,t}$, and let $\mu_{i,n} = \mathbb{E}[\bar{X}_{i,n}]$ be its expectation. Suppose $\mu_{i,n} \to \mu_i$ as $n \to \infty$ for all $i \in [K]$ and let

there exist constants, $\beta > 1$, $\xi > 0$, and $1/2 \le \eta < 1$ such that for every $z \ge 1$ and every integer $n \ge 1$:

$$\mathbb{P}(|\, n\bar{X}_{i,n} - n\mu_i \,| \ge n^{\eta} z) \le \frac{\beta}{z^{\xi}}.$$

For i.i.d. bounded rewards, the previous holds for $\eta = 1/2$ for any finite ξ because of exponential concentration. We propose to use the UCB algorithm where at time t, the arm I_t is chosen according to

$$I_t \in \arg\max_{i \in [K]} \{\bar{X}_{i,T_i(t-1)} + B_{t-1,T_i(t-1)}\}, \tag{1}$$

where $T_i(t) = \sum_{l=1}^{t} \mathbb{I}\{I_l = i\}$ is the number of times arm i has been played, up to (including) time t, and the bias or bonus term $B_{t,s}$ is defined as

$$B_{t,s} = \frac{\beta^{1/\xi} \cdot t^{\eta(1-\eta)}}{s^{1-\eta}}.$$

Let $\mu_* = \max_{i \in [K]} \mu_i$ and let $\bar{X}_n$ denote the empirical average of the rewards collected. Then, we establish that $\mathbb{E}[\bar{X}_n]$ converges to μ_*, and that for every $n \ge 1$ and every $z \ge 1$, a similar polynomial concentration holds:

$$\mathbb{P}(|n\bar{X}_n - n\mu_*| \ge n^{\eta} z) \le \frac{\beta'}{z^{\xi'}},$$

where $\xi' = \xi\eta(1-\eta) - 1$, and $\beta' > 1$ is a large enough constant. The precise statement can be found as Theorem 3 in Section 5.

1.1.2. Corrected UCT for MCTS and Nonasymptotic Analysis.

For MCTS, as discussed previously, the leaf nodes have rewards that can be viewed as generated per standard stationary MAB. Therefore, the rewards for each arm (or action) at the leaf level in MCTS satisfy the required concentration property with $\eta = 1/2$ because of independence. Hence, from our result for nonstationary MAB, we immediately obtain that we can recursively apply the UCB algorithm per (1) at each level in the MCTS with $\eta = 1/2$ and appropriately adjusted constants β and ξ. In effect, we obtain a modified UCT where the bias or bonus term $B_{t,s}$ scales as $t^{1/4}/s^{1/2}$. This is in constrast to $B_{t,s}$ scaling as $\sqrt{\log t/s}$ in the standard UCB and UCT suggested in the literature (Kocsis and Szepesvári 2006, Kocsis et al. 2006).

By recursively applying the convergence and concentration property of the nonstationary MAB for the resulting algorithm for MCTS, we establish that for any query state s of the MDP, using n simulations of the MCTS, we can obtain a value function estimation within error $\delta\varepsilon_0 + O(n^{-1/2})$, if we start with a value function estimation for all the leaf nodes within error ε_0 for some $\delta < 1$ (independent of n, dependent on depth of MCTS tree). That is, MCTS is indeed asymptotically correct as was conjectured in the prior literature. For details, see Theorem 1 in Section 3.

1.1.3. MCTS with Supervised Learning, Strong Policy Improvement, and Near Optimality.

The result stated previously for MCTS implies its "bootstrapping" property: if we start with a value function estimation for *all* state within error ε, then MCTS can produce estimation of value function for a *given query* state within error less than ε with enough simulations. By coupling such improved estimations of value function for a number of query states, combined with expressive enough supervised learning, one can hope to generalize such improved estimations of value function for *all* states. That is, MCTS coupled with supervised learning can be "strong policy improvement operator."

Indeed, this is precisely what we establish by using nearest neighbor supervised learning. Specifically, we establish that with $\tilde{O}(1/\varepsilon^{(4+d)})$[1] number of samples, MCTS with nearest neighbor finds an ε approximation of the optimal value function with respect to ℓ_∞-norm; here, d is the dimension of the state space. This is nearly optimal in view of a minimax lower bound of $\tilde{\Omega}(1/\varepsilon^{(2+d)})$ (Shah and Xie 2018). For details, see Theorem 2 in Section 4.

1.1.4. An Implication.

As mentioned earlier, the modified UCT policy per our result suggests using bias or bonus term $B_{t,s}$ that scales as $t^{1/4}/s^{1/2}$ at each node within the MCTS. Interestingly enough, the empirical results of AGZ are obtained by using $B_{t,s}$ that scales as $t^{1/2}/s$. This is qualitatively similar to what our results suggest and in contrast to the classical UCT.

1.2. Related Work

Reinforcement learning aims to approximate the optimal value function and policy directly from experimental data. A variety of algorithms have been developed, including model-based approaches, model-free approaches like tabular Q-learning (Watkins and Dayan 1992), and parametric approximation such as linear architectures (Sutton 1988). More recent work approximates the value function/policy by deep neural networks (Mnih et al. 2015; Schulman et al. 2015, 2017; Silver et al. 2017b; Yang et al. 2019), which can be trained using temporal-difference learning or Q-learning (Mnih et al. 2013, 2016; Van Hasselt et al. 2016).

MCTS is an alternative approach, which as discussed, estimates the (optimal) value of states by building a search tree from Monte Carlo simulations (Chang et al. 2005, Coulom 2006, Kocsis and Szepesvári 2006, Browne et al. 2012). Kocsis and Szepesvári (2006) and Kocsis et al. (2006) argue for the asymptotic convergence of MCTS with the standard UCT. However, the proof is incomplete. A key step toward proving the claimed result is to show the convergence and concentration properties of the regret for UCB under nonstationary reward distributions. In particular, to

establish an exponential concentration of regret (theorem 5 in Kocsis et al. 2006), Lemma 14 is applied. However, it requires conditional independence of $\{Z_i\}$ sequence, which does not hold, hence making the conclusion of exponential concentration questionable. Therefore, the proof of the main result (theorem 7 of Kocsis et al. 2006), which applies Theorem 5 with an inductive argument, is incorrect as stated.

In fact, it may be infeasible to prove theorem 5 in Kocsis et al. (2006) as it was stated. For example, the work of Audibert et al. (2009) shows that for bandit problems, the regret under UCB concentrates around its expectation polynomially rather than *exponentially* as desired in Kocsis et al. (2006) (e.g., if the essential infimum of the optimal arm's reward is below the mean reward of the second-best arm, see theorem 10 of Audibert et al. (2009)). Furthermore, Salomon and Audibert (2011) prove that for any strategy that does not use the knowledge of time horizon, it is infeasible to improve this polynomial concentration and establish exponential concentration. Our result is consistent with these fundamental bound of stationary MAB—we establish polynomial concentration of regret for nonstationary MAB, which plays a crucial role in our analysis of MCTS. Also see the work Munos (2014) for a discussion of the issues with logarithmic bonus terms for tree search.

Although we focus on UCT in this paper, we note that there are other variants of MCTS developed for a diverse range of applications. The work of Coquelin and Munos (2007) introduces flat UCB to improve the worst-case regret bounds of UCT. Schadd et al. (2008) modifies MCTS for single-player games by adding to the standard UCB formula a term that captures the possible deviation of the node. In the work by Sturtevant (2008), a variant of MCTS is introduced for multiplayer games by adopting the $\max^n$ idea. In addition to turn-based games like Go and Chess, MCTS has also been applied to real-time games (e.g., Ms. Pac-Man, Tron, and Starcraft) and nondeterministic games with imperfect information. The applications of MCTS go beyond games and appear in areas such as optimization, scheduling, and other decision-making problems. We refer to the survey on MCTS by Browne et al. (2012) for other variations and applications.

It has become popular recently to combine MCTS with deep neural networks, which serve to approximate the value function and/or policy (Silver et al. 2016, 2017a, b). For instance, in AGZ, MCTS uses the neural network to query the value of leaf nodes for simulation guidance; the neural network is then updated with sample data generated by MCTS-based policy and used in tree search in the next iteration. Azizzadenesheli et al. (2018) develop generative adversarial tree search that generates rollouts with a learned generative adversarial network–based dynamic model and reward predictor while using MCTS for planning over the simulated samples and a deep Q-network to query the Q-value of leaf nodes.

In terms of theoretical results, the closest work to our paper is Jiang et al. (2018), where they also consider a batch, MCTS-based reinforcement learning algorithm, which is a variant of the AGZ algorithm. The key algorithmic difference from ours lies in the leaf-node evaluator of the search tree: they use a combination of an estimated value function and an estimated policy. The latest observations at the root node are then used to update the value and policy functions (leaf-node evaluator) for the next iteration. They also give a finite sample analysis. However, their result and ours are quite different: in their analysis, the sample complexity of MCTS and the approximation power of value/policy architectures are *imposed as an assumption*; here we *prove* an explicit finite-sample bound for MCTS and characterize the nonasymptotic error prorogation under MCTS with nonparametric regression for leaf-node evaluation. Therefore, they *do not* establish "strong policy improvement" property of the MCTS.

Two other closely related papers are Teraoka et al. (2014) and Kaufmann and Koolen (2017), which study a simplified MCTS for two-player zero-sum games. There, the goal is to identify the best action of the root in a *given* game tree. For each leaf node, a stochastic oracle is provided to generate i.i.d. samples for the *true* reward. Teraoka et al. (2014) give a high probability bound on the number of oracle calls needed for obtaining ε-accurate score at the root. The more recent paper (Kaufmann and Koolen 2017) develops refined, instance-dependent sample complexity bounds. Compared with classical MCTS (e.g., UCT), both the setting and the algorithms in these papers are simpler: the game tree is given in advance rather than being built gradually through samples; the algorithm proposed in Teraoka et al. (2014) operates on the tree in a bottom-up fashion with uniform sampling at the leaf nodes. As a result, the analysis is significantly simpler and it is unclear whether the techniques can be extended to analyze other variants of MCTS.

It is important to mention the work of Chang et al. (2005) that explores the idea of using UCB for adaptive sampling in MDPs. The approximate value computed by the algorithm is shown to converge to the optimal value. We remark that their algorithm is different from the algorithm we analyze in this paper. In particular, their algorithm proceeds in a depth-first, recursive manner, and hence involves using UCB for a stationary MAB at each node. In contrast, the UCT algorithm we study involves nonstationary MABs; hence, our analysis is significantly different from theirs. We refer the readers to the work by Kocsis and Szepesvári (2006) and Coulom (2006) for further discussion of the difference. Another related work by Kearns et al. (2002)

studies a sparse sampling algorithm for large MDPs. This algorithm is also different from the MCTS family we analyze in this paper. Relatedly, Auger et al. (2013) consider a setting with finite horizon and continuous action space. During the tree simulation, a progressive widening technique is used to decide when to sample (add) a new action at each step; if no new action is needed for the current step, UCT is then extended with a *specific* choice and parameter of polynomial bonus for action selection. In contrast, we consider an infinite horizon setting. More importantly, we establish guarantees for a *class* of polynomial bonus forms determined by the set of interdependent algorithmic parameters. Again, this is made possible by introducing an appropriate form of nonstationary MAB, which could be of independent interest. Recently, this idea is further extended by Mao et al. (2020) to establish results for MCTS with a continuous armed bandit strategy, which shows more favorable performance than the algorithm proposed by Auger et al. (2013). We remark that the work by Efroni et al. (2018) studies multiple-step lookahead policies in reinforcement learning, which can be implemented via MCTS.

1.3. Organization

Section 2 describes the setting of MDP considered in this work. Section 3 describes the MCTS algorithm and the main result about its nonasymptotic analysis. Section 4 describes a reinforcement learning method that combines the MCTS with nearest neighbor supervised learning. It describes the finite-sample analysis of the method for finding ε approximate value function with respect to ℓ_∞ norm. Section 5 introduces a form of nonstationary multiarm bandit and an upper confidence bound policy for it. For this setting, we present the concentration of induced regret that serves as a key result for establishing the property of MCTS. The proofs of all the technical results are delegated to Sections 6–8 and the Appendices.

2. Setup and Problem Statement

2.1. Formal Setup

We consider the setup of the discrete-time discounted MDP. An MDP is described by a five-tuple $(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathcal{P}\equiv\mathcal{P}(s'\mid s,a)$ is the Markovian transition kernel, $\mathcal{R}\equiv\mathcal{R}(s,a)$ is a random reward function, and $\gamma\in(0,1)$ is a discount factor. At each time step, the system is in some state $s\in\mathcal{S}$. When an action $a\in\mathcal{A}$ is taken, the state transits to a next state $s'\in\mathcal{S}$ according to the transition kernel $\mathcal{P}$ and an immediate reward is generated according to $\mathcal{R}(s,a)$.

We consider the setup with access to the generative model (i.e., a simulator) (Kakade 2003), which is a common setting in the theoretical reinforcement learning literature. We assume that the agent has knowledge of $\mathcal{S}$, $\mathcal{A}$ and γ. The transition kernel $\mathcal{P}$ and the rewards $\mathcal{R}$ are unknown, but the agent could query the generative model at any given state-action pair (s,a) to obtain a sample of next state and the associated immediate reward.

A stationary policy $\pi(a\mid s)$ gives the probability of performing action $a\in\mathcal{A}$ given the current state $s\in\mathcal{S}$. The *value* function for each state $s\in\mathcal{S}$ under policy π, denoted by $V^\pi(s)$, is defined as the expected discounted sum of rewards received following the policy π from initial state s, that is,

$$V^\pi(s)=\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t\mathcal{R}(s_t,a_t)\mid s_0=s\right].$$

The goal is to find an optimal policy π^* that maximizes the value from each initial state. The optimal value function V^* is defined as $V^*(s)=V^{\pi^*}(s)=\sup_\pi V^\pi(s)$, $\forall s\in\mathcal{S}$. It is well understood that such an optimal policy exists in reasonable generality. In this paper, we restrict our attention to the MDPs with the following assumptions.

Assumption 1 (MDP Regularity). (A1) *The action space $\mathcal{A}$ is a finite set, and the state space $\mathcal{S}$ is a compact subset of a d-dimensional set; without loss of generality, let $\mathcal{S}=[0,1]^d$.* (A2) *The immediate rewards are random variables, uniformly bounded such that $\mathcal{R}(s,a)\in[-R_{\max},R_{\max}]$, $\forall s\in\mathcal{S},a\in\mathcal{A}$ for some $R_{\max}>0$.* (A3) *The state transitions are deterministic, that is, $\mathcal{P}\equiv\mathcal{P}(s'\mid s,a)\in\{0,1\}$ for all $s,s'\in\mathcal{S}$, $a\in\mathcal{A}$.*

Define $\beta\triangleq 1/(1-\gamma)$ and $V_{\max}\triangleq\beta R_{\max}$. Because all the rewards are bounded by $R_{\max}$, it is easy to see that the absolute value of the value function for any state under any policy is bounded by $V_{\max}$ (Even-Dar et al. 2003, Strehl et al. 2006).

2.1.1. On Deterministic Transition. We first remark that the deterministic transition in MDP is not a very restrictive assumption. Traditional artificial intelligence (AI) game research has been focused on deterministic games with a tree representation. MCTS has been extensively used in such deterministic transition problems (Browne et al. 2012), as demonstrated by the recent successes of MCTS in Go (Silver et al. 2017b), Chess (Silver et al. 2017a), and Atari games (Guo et al. 2014). There has been extensive theoretical literature on the analysis of MCTS and related methods for deterministic transitions (Hren and Munos 2008, Browne et al. 2012, Munos 2014, Bartlett et al. 2019), which provide crucial insights for more general scenarios in reinforcement learning.

Having noted that, our analysis and results for deterministic transitions indeed naturally extend to the stochastic setting with minor modifications. Considering the importance of deterministic transition setting and the clarity of our proof framework, we first

develop the results and the associated analysis for the setting of deterministic transitions. After presenting the main ideas, we shall extend them for the stochastic setting as described in Appendix A.

2.2. Value Function Iteration

A classical approach to find optimal value function, V^*, is an iterative approach called value function iteration. The Bellman equation characterizes the optimal value function as

$$V^*(s) = \max_{a\in\mathcal{A}}(\mathbb{E}[\mathcal{R}(s,a)] + \gamma V^*(s\circ a)), \tag{2}$$

where $s\circ a \in \mathcal{S}$ is the notation to denote the state reached by applying action a on state s. Under Assumption 1, the transitions are deterministic, and hence $s\circ a$ represents a single, deterministic state rather than a random state.

The value function iteration effectively views (2) as a fixed-point equation and tries to find a solution to it through a natural iteration. Precisely, let $V^{(t)}(\cdot)$ be the value function estimation in iteration t with $V^{(0)}$ being arbitrarily initialized. Then, for $t \geq 0$, for all $s \in \mathcal{S}$,

$$V^{(t+1)}(s) = \max_{a\in\mathcal{A}}(\mathbb{E}[\mathcal{R}(s,a)] + \gamma V^{(t)}(s\circ a)). \tag{3}$$

It is well known (Bertsekas 2017) that value iteration is contractive with respect to $\|\cdot\|_\infty$ norm for all $\gamma < 1$. Specifically, for $t \geq 0$, we have

$$\|V^{(t+1)} - V^*\|_\infty \leq \gamma \|V^{(t)} - V^*\|_\infty. \tag{4}$$

3. MCTS

The MCTS has been quite popular recently in many of reinforcement learning tasks. In effect, given a state $s \in \mathcal{S}$ and a value function estimate $\hat{V}$, it attempts to run the value function iteration for a fixed number of steps, say H, to evaluate $V^{(H)}(s)$ starting with $V^{(0)} = \hat{V}$ per (3). This, according to (4), would provide an estimate within error $\gamma^H \|\hat{V} - V^*\|_\infty$: an excellent estimate of $V^*(s)$ if H is large enough. The goal is to perform computation for value function iteration necessary to evaluate $V^{(H)}$ for state s only and not necessarily for all states as required by traditional value function iteration. MCTS achieves this by simply unrolling the associated computation tree. Another challenge that MCTS overcomes is the fact that value function iteration as in (3) assumes knowledge of model so that it can compute $\mathbb{E}[\mathcal{R}(\cdot,\cdot)]$ for any state-action pair. However, in reality, rewards are observed through samples and not a direct access to $\mathbb{E}[\mathcal{R}(\cdot,\cdot)]$. MCTS tries to use the samples in a careful manner to obtain accurate estimation for $V^{(H)}(s)$ over the computation tree suggested by the value function iteration as discussed previously. The concern of careful use of samples naturally connects it to MAB-like setting.

Next, we present a detailed description of the MCTS algorithm in Section 3.1. This can be viewed as a *correction* of the algorithm presented in Kocsis and Szepesvári (2006) and Kocsis et al. (2006). We state its theoretical property in Section 3.2.

3.1. Algorithm

We provide details of a specific form of MCTS, which replaces the logarithmic bonus term of UCT with a polynomial one. Overall, we fix the search tree to be of depth H. Similar to most literature on this topic, it uses a variant of the UCB algorithm to select an action at each stage. At a leaf node (i.e., a state at depth H), we use the current value oracle $\hat{V}$ to evaluate its value. Because we consider deterministic transitions, consequently, the tree is fixed once the root node (state) is chosen, and we use the notation $s\circ a$ to denote the next state after taking action a at state s. Each edge represents a state-action pair, whereas each node represents a state. For clarity, we use superscript to distinguish quantities related to different depth. The pseudo-code for the MCTS procedure is given in Algorithm 1, and Figure 1 shows the structure of the search tree and related notation.

Algorithm 1 (Fixed-Depth MCTS)

Input: (1) current value oracle $\hat{V}$, root node $s^{(0)}$ and search depth H;
(2) number of MCTS simulations n;
(3) algorithmic constants, $\{\alpha^{(i)}\}_{i=1}^H$, $\{\beta^{(i)}\}_{i=1}^H$, $\{\xi^{(i)}\}_{i=1}^H$ and $\{\eta^{(i)}\}_{i=1}^H$.

Initialization: for each depth h, initialize the cumulative node value $\tilde{v}^{(h)}(s) = 0$ and visit count $N^{(h)}(s) = 0$ for every node s and initialize the cumulative edge value $q^{(h)}(s,a) = 0$.

for each MCTS simulation $t = 1, 2, \ldots, n$ **do**

`/* Simulation: select actions until reaching depth H*/`

for depth $h = 0, 1, 2, \ldots, H-1$ **do**

at state $s^{(h)}$ of depth h, select an action (edge) according to

$$a^{(h+1)} = \arg\max_{a\in\mathcal{A}} \frac{q^{(h+1)}(s^{(h)},a) + \gamma\tilde{v}^{(h+1)}(s^{(h)}\circ a)}{N^{(h+1)}(s^{(h)}\circ a)} + \frac{(\beta^{(h+1)})^{1/\xi^{(h+1)}} \cdot (N^{(h)}(s^{(h)}))^{\alpha^{(h+1)}/\xi^{(h+1)}}}{(N^{(h+1)}(s^{(h)}\circ a))^{1-\eta^{(h+1)}}}, \tag{5}$$

where dividing by zero is assumed to be $+\infty$.

upon taking the action $a^{(h+1)}$, receive a random reward $r^{(h+1)} \triangleq \mathcal{R}(s^{(h)}, a^{(h+1)})$ and transit to a new state $s^{(h+1)}$ at depth $h+1$.

end for

`/* Evaluation: call value oracle for leaf nodes*/`

10: reach $s^{(H)}$ at depth H, call the current value oracle and let $\tilde{v}^{(H)}(s^{(H)}) = \hat{V}(s^{(H)})$.
11: /* Update Statistics: quantities on the search path */
12: **for** depth $h = 0, 1, 2, \ldots, H-1$ **do**
13: update statistics of nodes and edges that are on the search path of current simulation:

$$\text{visit count} : N^{(h+1)}(s^{(h+1)}) = N^{(h+1)}(s^{(h+1)}) + 1$$

$$\text{edge value} : q^{(h+1)}(s^{(h)}, a^{(h+1)}) = q^{(h+1)}(s^{(h)}, a^{(h+1)}) + r^{(h+1)}$$

$$\text{node value} : \tilde{v}^{(h)}(s^{(h)}) = \tilde{v}^{(h)}(s^{(h)}) + r^{(h+1)} + \gamma r^{(h+2)} + \cdots + \gamma^{H-1-h} r^{(H)} + \gamma^{H-h} \tilde{v}^{(H)}(s^{(H)})$$

14: **end for**
15: **end for**
16: **Output:** average of the value for the root node $\tilde{v}^{(0)}(s^{(0)})/n$.

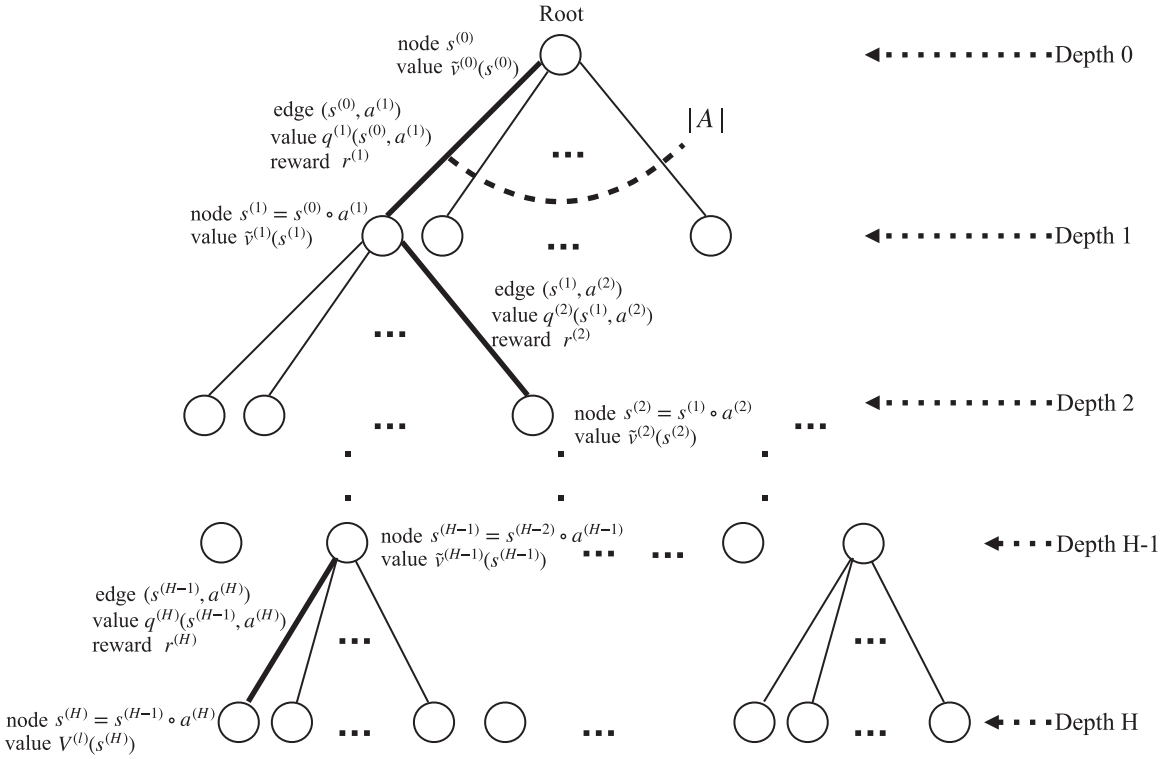
In Algorithm 1, there are certain sequences of algorithmic parameters required, namely, α, β, ξ, and η. The choices for these constants will become clear in our nonasymptotic analysis. At a higher level, the constants for the last layer (i.e., depth H), $\alpha^{(H)}$, $\beta^{(H)}$, $\xi^{(H)}$ and $\eta^{(H)}$ depend on the properties of the leaf nodes, whereas the rest are recursively determined by the constants one layer below. We note that in selecting action $a^{(h+1)}$ at each depth h (i.e., Line 6 of Algorithm 1), the upper confidence term is polynomial in n, whereas a typical UCB algorithm would be logarithmic in n, where n is the number of visits to the corresponding state thus far. The logarithmic factor in the original UCB algorithm was motivated by the exponential tail probability bounds. In our case, it turns out that exponential tail bounds for each layer seems to be infeasible without further structural assumptions. As mentioned in Section 1.2, prior work (Audibert et al. 2009, Salomon and Audibert 2011) has justified the polynomial concentration of the regret for the classical UCB in the stochastic (independent rewards) MAB setting. This implies that the concentration at intermediate depth (i.e., depth less than H) is at most polynomial. Indeed, we will prove these polynomial concentration bounds even for the nonstationary (dependent, nonstationary rewards) MAB that shows up in MCTS and discuss separately in Section 5.

3.2. Analysis

Now, we state the following result on the nonasymptotic performance of the MCTS as described previously.

Theorem 1. *Consider an MDP satisfying Assumption 1. Let $H \geq 1$, and for $1/2 \leq \eta < 1$, let*

Figure 1. Notation and Sample Simulation Path of MCTS (Thick Lines)



$$\eta^{(h)} = \eta^{(H)} \equiv \eta, \qquad \forall h \in [H], \tag{6}$$

$$\alpha^{(h)} = \eta(1-\eta)(\alpha^{(h+1)} - 1), \qquad \forall h \in [H-1], \tag{7}$$

$$\xi^{(h)} = \alpha^{(h+1)} - 1, \qquad \forall h \in [H-1]. \tag{8}$$

Suppose that a large enough $\xi^{(H)}$ is chosen such that $\alpha^{(1)} > 2$. Then, there exist corresponding constants $\{\beta^{(i)}\}_{i=1}^{H}$ such that for each query state $s \in \mathcal{S}$, the following claim holds for the output $\hat{V}_n(s)$ of MCTS with n simulations:

$$|\mathbb{E}[\hat{V}_n(s)] - V^*(s)| \le \gamma^H \varepsilon_0 + O(n^{\eta-1}), \tag{9}$$

where $\varepsilon_0 = \|\hat{V} - V^*\|_\infty$ with $\hat{V}$ being the estimate of V^* used by the MCTS algorithm for leaf nodes.

Because $\eta \in [1/2, 1)$, Theorem 1 implies a best case convergence rate of $O(n^{-1/2})$ by setting $\eta = 1/2$. The constant in the $O(\cdot)$ notation also depends on $\eta \in [\frac{1}{2}, 1)$. However, the impact of η on n is entirely captured through $n^{\eta-1}$. Therefore, the order-wise optimal convergence is achieved by the choice of $\eta = 1/2$. With these parameter choices, the bias term in the upper confidence bound (Line 6 of Algorithm 1) scales as $(N^{(h)}(s^{(h)}))^{1/4}/\sqrt{N^{(h+1)}(s^{(h)} \circ a)}$, that is, in the form of $t^{1/4}/\sqrt{S}$ as mentioned in Section 1, where $t \equiv N^{(h)}(s^{(h)})$ is the number of times that state $s^{(h)}$ at depth h has been visited, and $S \equiv N^{(h+1)}(s^{(h)} \circ a)$ is the number of times action a has been selected at state $s^{(h)}$.

3.2.1. High Probability Bound. Theorem 1 states bounds on expected estimation error in value function (cf. (9)). We remark that the proof is established via recursively arguing a certain form of convergence and polynomial concentration properties for the nonstationary value function estimate sequence for nodes at each depth. That is, starting with the convergence and polynomial concentration properties for nodes at depth $h+1$, we establish a similar form of convergence and polynomial concentration properties for nodes at depth h. We recursively apply this argument, starting from the leaf nodes, until reaching the root node. Therefore, the output $\hat{V}_n(s)$ of MCTS at root node also satisfies a form of polynomial concentration. Specifically, under the setup of Theorem 1, it follows that for every $n \ge 1$ and every $z \ge 1$:

$$\mathbb{P}(n\hat{V}_n(s) - n\mu^*(s) \ge n^\eta z) \le \frac{\beta^{(1)}}{z^{\xi^{(0)}}}, \quad \mathbb{P}(n\hat{V}_n(s) - n\mu^*(s) \le -n^\eta z) \le \frac{\beta^{(1)}}{z^{\xi^{(0)}}},$$

where η, $\xi^{(0)}$, and $\beta^{(1)}$ are some constants (see Theorem 1 and the proof in Section 7 for details.). Here, $\mu^*(s)$ is the value function estimation for s after H iterations of value function iteration starting with $\hat{V}$. With the classical contraction result for value function iteration, that is, $|\mu^*(s) - V^*(s)| \le \gamma^H \varepsilon_0$, we obtain

$$\mathbb{P}(n\hat{V}_n(s) - nV^*(s) \ge n^\eta z + \gamma^H \varepsilon_0) \le \frac{\beta^{(1)}}{z^{\xi^{(0)}}},$$

$$\mathbb{P}(n\hat{V}_n(s) - nV^*(s) \le -n^\eta z - \gamma^H \varepsilon_0) \le \frac{\beta^{(1)}}{z^{\xi^{(0)}}}.$$

4. Reinforcement Learning Through MCTS with Supervised Learning

Recently, MCTS has been used prominently in various empirical successes of reinforcement learning including AGZ. Here, MCTS is combined with expressive supervised learning method to iteratively improve the policy and the value function estimation. In effect, MCTS combined with supervised learning acts as a policy improvement operator.

Intuitively, MCTS produces an improved estimation of value function for a given state of interest, starting with a given estimation of value function by unrolling the computation tree associated with value function iteration. MCTS achieves this using observations obtained through simulations. Establishing this improvement property rigorously was the primary goal of Section 3. Now, given such improved estimation of value function for finitely many states, a good supervised learning method can learn to generalize such an improvement to all states. If so, this is like performing value function iteration, but using simulations. Presenting such a policy and establishing such guarantees is the crux of this section.

To that end, we present a reinforcement learning method that combines MCTS with nearest neighbor supervised learning. For this method, we establish that indeed, with sufficient number of samples, the resulting policy improves the value function estimation just like value function iteration. Using this, we provide a finite-sample analysis for learning the optimal value function within a given tolerance. We find it nearly matching a minimax lower bound in Shah and Xie (2018), which we recall in Section 4.4, and thus establishes near minimax optimality of such a reinforcement learning method.

4.1. Reinforcement Learning Policy

Here we describe the policy to produce estimation of optimal value function V^*. Similar approach can be applied to obtain estimation of policy as well. Let $V^{(0)}$ be the initial estimation of V^*, and for simplicity, let $V^{(0)}(\cdot) = 0$. We describe a policy that iterates between use of MCTS and supervised learning to iteratively obtain estimation $V^{(\ell)}$ for $\ell \ge 1$, so that iteratively better estimation of V^* is produced as ℓ increases. To that end, for $\ell \ge 1$:

- For appropriately sampled states $S^\ell = \{s_i\}_{i=1}^{m_\ell}$, apply MCTS to obtain improved estimations of value function $\{\hat{V}^{(\ell)}(s_i)\}_{i=1}^{m_\ell}$ using $V^{(\ell-1)}$ to evaluate leaf nodes during simulations.

- Using $\{(s_i, \hat{V}^{(\ell)}(s_i)\}_{i=1}^{m_\ell}$ with a variant of nearest neighbor supervised learning with parameter $\delta_\ell \in (0,1)$, produce estimation $V^{(\ell)}$ of the optimal value function.

For completeness, the pseudo-code is provided in Algorithm 2.

Algorithm 2 (Reinforcement Learning Policy)

Input: initial value function oracle $V^{(0)}(s) = 0$, $\forall s \in \mathcal{S}$
for $l = 1, 2, \ldots, L$ **do**
  `/* improvement via MCTS */`
  uniformly and independently sample states $S^\ell = \{s_i\}_{i=1}^{m_\ell}$.
  for each sampled state s_i **do**
    apply the MCTS algorithm, which takes as inputs the current value oracle $V^{(l-1)}$, the depth $H^{(l)}$, the number of simulation n_l, and the root node s_i, and outputs an improved estimate for $V^*(s_i)$:

$$\hat{V}^{(l)}(s_i) = \mathrm{MCTS}(V^{(l-1)}, H^{(l)}, n_l, s_i) \tag{10}$$

  end for
  `/* supervised learning */`
  with the collected data $\mathcal{D}^{(l)} = \{(s_i, \hat{V}^{(l)}(s_i))\}_{i=1}^{m_l}$, build a new value oracle $V^{(l)}$ via a nearest neighbor regression with parameter δl:

$$V^{(l)}(s) = \text{Nearest Neigbhor}\,(\mathcal{D}^{(l)}, \delta_l, s), \quad \forall s \in \mathcal{S}. \tag{11}$$

end for
Output: final value oracle $V^{(L)}$.

4.2. Supervised Learning

For simplicity, we shall use the following variant of the nearest neighbor supervised learning parametrized by $\delta \in (0,1)$. Given state space $\mathcal{S} = [0,1]^d$, we wish to cover it with minimal (up to scaling) number of balls of radius δ (with respect to ℓ_2-norm). To that end, because $\mathcal{S} = [0,1]^d$, one such construction is where we have balls of radius δ with centers being $\{(\theta_1, \theta_2, \ldots, \theta_d) : \theta_1, \ldots, \theta_d \in \mathcal{Q}(\delta)\}$ where

$$\mathcal{Q}(\delta) = \left\{\frac{1}{2}\delta i \ : \ i \in \mathbb{Z}, 0 \le i \le \left\lfloor \frac{2}{\delta} \right\rfloor \right\} \bigcup \left\{1 - \frac{1}{2}\delta i \ : \ i \in \mathbb{Z}, 0 \le i \le \left\lfloor \frac{2}{\delta} \right\rfloor \right\}.$$

Let the collection of these balls be denoted by $c_1, \ldots, c_{K(\delta,d)}$ with $K(\delta, d) = |\mathcal{Q}(\delta)|$. It is easy to verify that $\mathcal{S} \subset \bigcup_{i \in [K(\delta,d)]} c_i$, $K(\delta, d) = \Theta(\delta^{-d})$ and $C_d \delta^d \le \text{volume}\,(c_i \cap \mathcal{S}) \le C'_d \delta^d$ for strictly positive constants C_d, C'_d that depends on d but not δ. For any $s \in \mathcal{S}$, let $j(s) = \min\{j : s \in c_j\}$. Given observations $\{(s_i, \hat{V}^{(\ell)}(s_i)\}_{i=1}^{m_\ell}$, we produce an estimate $V^{(\ell)}(s)$ for all $s \in \mathcal{S}$ as follows:

$$V^{(\ell)}(s) = \begin{cases} \dfrac{\sum_{i: s_i \in c_{j(s)}} \hat{V}^{(\ell)}(s_i)}{|\{i : s_i \in c_{j(s)}\}|}, & \text{if } |\{i : s_i \in c_{j(s)}\}| \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

It is worth noting that other supervised learning algorithms could be used to achieve similar performance guarantees under proper conditions. In this work, as a concrete example, we instantiate the supervised learning algorithm with nearest neighbors for its simplicity and its generalization guarantee for smooth functions. As only the basic Lipschitz smoothness is assumed (Assumption 2 in Section 4.3), we do not expect order-wise gain in terms of improving sample complexity from other learning methods. However, it could be beneficial to use a more refined method, for example, local polynomial interpolation, if the underlying function posses higher-order smoothness or a parametric form.

4.3. Finite-Sample Analysis

For finite-sample analysis of the proposed reinforcement learning policy, we make the following structural assumption about the MDP. Specifically, we assume that the optimal value function (i.e., true regression function) is smooth in some sense. We note that some form of smoothness assumption for MDPs with continuous state/action space is typical for ℓ_∞ guarantee. The Lipschitz continuous assumption stated here is natural and representative in the literature on MDPs with continuous state spaces (Bertsekas 1975; Dufour and Prieto-Rumeau 2012, 2013; Munos 2014).

Assumption 2. (Smoothness). *The optimal value function $V^* : \mathcal{S} \to \mathbb{R}$ satisfies Lipschitz continuity with parameter C, that is, $\forall s, s' \in \mathcal{S} = [0,1]^d$, $|V^*(s) - V^*(s')| \le C \|s - s'\|_2$.*

Now we state the result characterizing the performance of the reinforcement learning policy described previously.

Theorem 2. *Let Assumptions 1 and 2 hold. Let $\varepsilon > 0$ be a given error tolerance. Then, for $L = \Theta\left(\log \frac{\varepsilon}{V_{\max}}\right)$, with appropriately chosen m_ℓ, δ_ℓ for $\ell \in [L]$, as well as parameters in MCTS, the reinforcement learning algorithm produces estimation of value function $V^{(L)}$ such that*

$$\mathbb{E}[\sup_{s \in \mathcal{S}} |V^{(L)}(s) - V^*(s)|] \le \varepsilon,$$

by selecting m_ℓ states uniformly at random in $\mathcal{S}$ within iteration ℓ. This, in total, requires T number of state transitions (or samples), where

$$T = O\left(\varepsilon^{-(4+d)} \cdot \left(\log \frac{1}{\varepsilon}\right)^5\right).$$

A few remarks are in order.We first note that, in Theorem 2, the expectation is taken with respect to all

the randomness in the algorithm, that is, the randomness in sampling the states at each iteration (Line 4 of Algorithm 2) and the randomness in each query of the MCTS algorithm (Line 6 of Algorithm 2). As for the parameters for MCTS and supervised learning, the following choice would lead to the guarantee of Theorem 2: for each iteration $\ell \geq 1$, we set

$$H^{(\ell)} = c_0' \log \lambda^{-1}, \quad n_\ell = c_2' \lambda^{-\ell/(1-\eta)}, \quad \delta_\ell = c_1' \lambda^{\ell} \quad \text{and}$$
$$m_\ell = c_3' \delta_\ell^{-d-2} \log \delta_\ell^{-1},$$

where $\lambda \triangleq (\varepsilon / V_{\max})^{1/L}$, and c_0', c_1', c_2', c_3' are constants independent of λ and ℓ; see Section 8.2 for details.

4.4. Minimax Lower Bound

Leveraging the minimax lower bound for the problem of nonparametric regression (Stone 1982, Tsybakov 2009), recent work (Shah and Xie 2018) establishes a lower bound on the sample complexity for reinforcement learning algorithms for general MDPs without additional structural assumptions. Indeed the lower bound also holds for MDPs with deterministic transitions (the proof is provided in Appendix C), which is stated in the following proposition.

Proposition 1. *Given an algorithm, let V_T be the estimation of V^* after T samples of state transitions for the given MDP. Then, for each $\varepsilon \in (0,1)$, there exists an instance of deterministic MDP such that to achieve $\mathbb{P}[\| \hat{V}_T - V^* \|_\infty < \varepsilon] \geq 1 - \varepsilon$, it must be that*

$$T \geq C' d \cdot \varepsilon^{-(d+2)} \cdot \log(\varepsilon^{-1}),$$

where $C' > 0$ is a constant independent of the algorithm.

Proposition 1 states that for any policy to learn the optimal value function within ε approximation error, the number of samples required must scale as $\tilde{\Omega}(\varepsilon^{-(2+d)})$. Theorem 2 implies that the sample complexity of the proposed algorithm scales as $\tilde{O}(\varepsilon^{-(4+d)})$ (omitting the logarithmic factor). Hence, in terms of the dependence on the dimension, the proposed algorithm is nearly optimal. Optimizing the dependence of the sample complexity on other parameters is an important direction for future work.

5. Nonstationary MAB

We introduce a class of nonstationary MAB problems, which will play a crucial role in analyzing the MCTS algorithm. To that end, let there be $K \geq 1$ arms or actions of interest. Let $X_{i,t}$ denote the random reward obtained by playing the arm $i \in [K]$ for the tth time with $t \geq 1$. Let $\bar{X}_{i,n} = \frac{1}{n} \sum_{t=1}^{n} X_{i,t}$ denote the empirical average of playing arm i for n times, and let $\mu_{i,n} = \mathbb{E}[\bar{X}_{i,n}]$ be its expectation. For each arm $i \in [K]$, the reward $X_{i,t}$ is bounded in $[-R, R]$ for some $R > 0$, and we assume that the reward sequence, $\{X_{i,t} : t \geq 1\}$, is a nonstationary process satisfying the following convergence and concentration properties:

A. Convergence: The expectation $\mu_{i,n}$ converges to a value μ_i, that is,

$$\mu_i = \lim_{n \to \infty} \mathbb{E}[\bar{X}_{i,n}]. \tag{13}$$

B. Concentration: There exist three constants, $\beta > 1$, $\xi > 0$, and $1/2 \leq \eta < 1$ such that for every $z \geq 1$ and every integer $n \geq 1$,

$$\mathbb{P}(n\bar{X}_{i,n} - n\mu_i \geq n^\eta z) \leq \frac{\beta}{z^\xi}, \quad \mathbb{P}(n\bar{X}_{i,n} - n\mu_i \leq -n^\eta z) \leq \frac{\beta}{z^\xi}. \tag{14}$$

5.1. Algorithm

Consider applying the following variant of the UCB algorithm to the nonstationary MAB. Define the UCB for arm or action i when it is played s times in total of $t \geq s$ time steps as

$$U_{i,s,t} = \bar{X}_{i,s} + B_{t,s}, \tag{15}$$

where $B_{t,s}$ is the "bonus term." Denote by I_t the arm played at time $t \geq 1$. Then,

$$I_t \in \arg\max_{i \in [K]} \{\bar{X}_{i,T_i(t-1)} + B_{t-1,T_i(t-1)}\}, \tag{16}$$

where $T_i(t) = \sum_{l=1}^{t} \mathbb{I}\{I_l = i\}$ is the number of times arm i has been played, up to (including) time t. We shall make specific selection of the bonus or bias term $B_{t,s}$ as

$$B_{t,s} = \frac{\beta^{1/\xi} \cdot t^{\alpha/\xi}}{s^{1-\eta}}. \tag{17}$$

A tie is broken arbitrarily when selecting an arm. In the previous statements, $\alpha > 0$ is a tuning parameter that controls the exploration and exploitation tradeoff. Let $\mu_* = \max_{i \in [K]} \mu_i$ be the optimal value with respect to the converged expectation and $i_* \in \arg\max_{i \in [K]} \mu_i$ be the corresponding optimal arm. We assume that the optimal arm is unique. Let $\delta_{i*,n} = \mu_{i_*,n} - \mu_{i_*}$, which measures how fast the mean of the optimal nonstationary arm converges. For simplicity, quantities related to the optimal arm i_* will be simply denoted with subscript $*$, for example, $\delta_{*,n} = \delta_{i_*,n}$. Finally, denote by $\Delta_{\min} = \min_{i \in [K], i \neq i_*} \Delta_i$ the gap between the optimal arm and the second optimal arm with notation $\Delta_i = \mu_* - \mu_i$.

5.2. Analysis

Let $\bar{X}_n \triangleq \frac{1}{n} \sum_{i=1}^{K} T_i(n) \bar{X}_{i,T_i(n)}$ denote the empirical average under the UCB algorithm (16). Then, $\bar{X}_n$ satisfies the following convergence and concentration properties.

Theorem 3. *Consider a nonstationary MAB satisfying (13) and (14). Suppose that Algorithm (16) is applied with*

parameter α such that $\xi\eta(1-\eta) \le \alpha < \xi(1-\eta)$ and $\alpha > 2$. Then, the following holds:

A. *Convergence*:

$$|\mathbb{E}[\bar{X}_n] - \mu_*| \le |\delta_{*,n}| + \frac{2R(K-1)\cdot\left(\left(\frac{2}{\Delta_{\min}}\cdot\beta^{1/\xi}\right)^{\frac{1}{1-\eta}}\cdot n^{\frac{\alpha}{\xi(1-\eta)}} + \frac{2}{\alpha-2} + 1\right)}{n}.$$

B. *Concentration: there exist constants, $\beta' > 1$ and $\xi' > 0$ and $1/2 \le \eta' < 1$ such that for every $n \ge 1$ and every $z \ge 1$,*

$$\mathbb{P}(n\bar{X}_n - n\mu_* \ge n^{\eta'}z) \le \frac{\beta'}{z^{\xi}}, \quad \mathbb{P}(n\bar{X}_n - n\mu_* \le -n^{\eta'}z) \le \frac{\beta'}{z^{\xi'}},$$

where $\eta' = \frac{\alpha}{\xi(1-\eta)}$, $\xi' = \alpha - 1$, β' depends on $R, K, \Delta_{\min}, \beta, \xi, \alpha, \eta$.

6. Proof of Theorem 3

We establish the convergence and concentration properties of the variant of the UCB algorithm described in Section 5 and specified through (15)–(17).

Establishing the Convergence Property. We define a useful notation

$$\Phi(n,\delta) = n^{\eta}\left(\frac{\beta}{\delta}\right)^{1/\xi}. \tag{18}$$

We begin with a useful lemma, which shows that the probability that a nonoptimal arm or action has a large upper confidence is polynomially small. Proof is provided in Section 6.1.

Lemma 1. *Let $i \in [K], i \ne i_*$ be a suboptimal arm and define*

$$A_i(t) \triangleq \min_{u\in\mathbb{N}}\left\{\frac{\Phi(u,t^{-\alpha})}{u} \le \frac{\Delta_i}{2}\right\} = \left\lceil\left(\frac{2}{\Delta_i}\cdot\beta^{1/\xi}\cdot t^{\alpha/\xi}\right)^{\frac{1}{1-\eta}}\right\rceil. \tag{19}$$

For each s and t such that, $A_i(t) \le s \le t$, we have

$$\mathbb{P}(U_{i,s,t} > \mu_*) \le t^{-\alpha}.$$

Lemma 1 implies that as long as each arm is played enough, the suboptimal ones become less likely to be selected. This allows us to upper bound the expected number of suboptimal plays as follows.

Lemma 2. *Let $i \in [K], i \ne i_*$, then*

$$\mathbb{E}[T_i(n)] \le \left(\frac{2}{\Delta_i}\cdot\beta^{1/\xi}\right)^{\frac{1}{1-\eta}}\cdot n^{\frac{\alpha}{\xi(1-\eta)}} + \frac{2}{\alpha-2} + 1.$$

The proof of Lemma 2 is deferred to Section 6.2.

Completing Proof of Convergence. By the triangle inequality,

$$|\mu_* - \mathbb{E}[\bar{X}_n]| = |\mu_* - \mu_{*,n}| + |\mu_{*,n} - \mathbb{E}[\bar{X}_n]| = |\delta_{*,n}| + |\mu_{*,n} - \mathbb{E}[\bar{X}_n]|.$$

The second term can be bounded as follows:

$$\begin{aligned} n|\mu_{*,n} - \mathbb{E}[\bar{X}_n]| &= \left|\mathbb{E}\left[\sum_{t=1}^{n} X_{i_*,t}\right] - \mathbb{E}\left[\sum_{i=1}^{K} T_i(n)\bar{X}_{i,T_i(n)}\right]\right| \\ &\le \left|\mathbb{E}\left[\sum_{t=1}^{n} X_{i_*,t}\right] - \mathbb{E}[T_*(n)\bar{X}_{i_*,T_*(n)}]\right| + \left|\mathbb{E}\left[\sum_{i=1,i\ne i_*}^{K} T_i(n)\bar{X}_{i,T_i(n)}\right]\right| \\ &= \left|\mathbb{E}\left[\sum_{t=T_*(n)+1}^{n} X_{i_*,t}\right]\right| + \left|\mathbb{E}\left[\sum_{i=1,i\ne i_*}^{K} T_i(n)\bar{X}_{i,T_i(n)}\right]\right|. \end{aligned} \tag{20}$$

Recall that the reward sequences are assumed to be bounded in $[-R,R]$. Therefore, the first term of (20) can be bounded as follows:

$$\left|\mathbb{E}\left[\sum_{t=T_*(n)+1}^{n} X_{i_*,t}\right]\right| \le \mathbb{E}\left[\sum_{t=T_*(n)+1}^{n} |X_{i_*,t}|\right] \le R\cdot\mathbb{E}\left[\sum_{i=1,i\ne i_*}^{K} T_i(n)\right].$$

The second term can also be bounded as

$$\left|\mathbb{E}\left[\sum_{i=1,i\ne i_*}^{K} T_i(n)\bar{X}_{i,T_i(n)}\right]\right| \le R\cdot\mathbb{E}\left[\sum_{i=1,i\ne i_*}^{K} T_i(n)\right].$$

Hence, we obtain that

$$|\mu_* - \mathbb{E}[\bar{X}_n]| = |\delta_{*,n}| + |\mu_{*,n} - \mathbb{E}[\bar{X}_n]| \le |\delta_{*,n}| + \frac{2R\cdot\mathbb{E}\left[\sum_{i=1,i\ne i_*}^{K} T_i(n)\right]}{n}.$$

Combining the above bounds and Lemma 2 yields the desired convergence result in Theorem 3.

Establishing the Concentration Property. Having proved the convergence property, the next step is to show that a similar concentration property (cf. (14)) also holds for $\bar{X}_n$. We aim to precisely capture the relationship between the original constants assumed in the assumption and the new constants obtained for $\bar{X}_n$. To begin with, recall the definition of $A_i(t)$ in Lemma 1 and define

$$A(t) = \max_{i\in[K]} A_i(t) = \left\lceil\left(\frac{2}{\Delta_{\min}}\cdot\beta^{1/\xi}\right)^{\frac{1}{1-\eta}}\cdot t^{\frac{\alpha}{\xi(1-\eta)}}\right\rceil. \tag{21}$$

It can be checked that replacing β with any larger number still makes the concentration inequalities (14) hold. Without loss of generality, we hence let β be large enough so that $\frac{2}{\Delta_{\min}}\cdot\beta^{1/\xi} > 1$. We further denote by N_p the first time such that $t \ge A(t)$, that is,

$$N_p = \min\{t \ge 1 : t \ge A(t)\} = \Theta\left(\left(\frac{2^{\xi}\beta}{\Delta_{\min}{}^{\xi}}\right)^{\frac{1}{\xi(1-\eta)-\alpha}}\right). \tag{22}$$

We first state the following concentration property, which will be further refined to match the desired form in Theorem 3. We defer the proof to Section 6.3.

Lemma 3. *For any $n \geq N_p$ and $x \geq 1$, let $r_0 = n^\eta + 2R(K-1)(3+A(n))$. Then,*

$$\mathbb{P}(n\bar{X}_n - n\mu_* \geq r_0 x) \leq \frac{\beta}{x^\xi} + \frac{2(K-1)}{(\alpha-1)((1+A(n))x)^{\alpha-1}},$$
$$\mathbb{P}(n\bar{X}_n - n\mu_* \leq -r_0 x) \leq \frac{\beta}{x^\xi} + \frac{2(K-1)}{(\alpha-1)((1+A(n))x)^{\alpha-1}}.$$

Lemma 3 confirms that indeed, as n becomes large, the average $\bar{X}_n$ also satisfies certain concentration inequalities. However, the particular form of concentration in Theorem 3 does not quite match the form of concentration in Theorem 3, which we conclude next.

Completing Proof of Concentration Property. Let N_p' be a constant defined as follows:

$$N_p' = \min\{t \geq 1 : t \geq A(t) \text{ and } 2RA(t) \geq t^\eta + 2R(4K-3)\}.$$

Recall the definition of $A(t)$ and that $\alpha \geq \xi\eta(1-\eta)$ and $\alpha < \xi(1-\eta)$. Hence, N_p' is guaranteed to exist. In addition, by definition, $N_p' \geq N_p$. For each $n \geq N_p'$,

$$\begin{aligned}
2RK\left(\tfrac{2}{\Delta_{\min}}\cdot\beta^{1/\xi}\right)^{\frac{1}{1-\eta}}\cdot n^{\frac{\alpha}{\xi(1-\eta)}} &= 2RK\left[\left(\tfrac{2}{\Delta_{\min}}\cdot\beta^{1/\xi}\right)^{\frac{1}{1-\eta}}\cdot n^{\frac{\alpha}{\xi(1-\eta)}} + 1 - 1\right] \\
&\geq 2RKA(n) - 2RK \\
&= 2R(K-1)A(n) + 2RA(n) - 2RK \\
&\geq 2R(K-1)A(n) + n^\eta \\
&\quad + 2R(4K-3) - 2RK \\
&= 2R(K-1)(A(n)+3) + n^\eta = r_0
\end{aligned}$$

Now, let us apply Lemma 3: for every $n \geq N_p'$ and $x \geq 1$, we have

$$\begin{aligned}
\mathbb{P}(n\bar{X}_n - n\mu_* \geq n^{\frac{\alpha}{\xi(1-\eta)}}\left[2RK\left(\tfrac{2}{\Delta_{\min}}\cdot\beta^{1/\xi}\right)^{\frac{1}{1-\eta}}\right]x) &\leq \mathbb{P}(n\bar{X}_n - n\mu_* \geq r_0 x) \\
&\leq \frac{\beta}{x^\xi} + \frac{2(K-1)}{(\alpha-1)((1+A(n))x)^{\alpha-1}} \\
&\leq \frac{2\max\left(\beta, \frac{2(K-1)}{(\alpha-1)(1+A(N_p'))^{\alpha-1}}\right)}{x^{\alpha-1}},
\end{aligned} \tag{23}$$

where the last inequality follows because $n \geq N_p'$ and $A(n)$ is a nondecreasing function. In addition, because $\alpha < \xi(1-\eta) < \xi$, we have $\alpha - 1 < \xi$. For convenience, we define a constant

$$c_1 \triangleq 2RK\left(\frac{2}{\Delta_{\min}}\cdot\beta^{1/\xi}\right)^{\frac{1}{1-\eta}}. \tag{24}$$

Equivalently, by a change of variable, that is, letting $z = c_1 x$, then for every $n \geq N_p'$ and $z \geq 1$, we obtain that

$$\mathbb{P}\left(n\bar{X}_n - n\mu_* \geq n^{\frac{\alpha}{\xi(1-\eta)}}z\right) \leq \frac{2c_1^{\alpha-1}\cdot\max\left(\beta, \frac{2(K-1)}{(\alpha-1)(1+A(N_p'))^{\alpha-1}}\right)}{z^{\alpha-1}}. \tag{25}$$

The previous inequality holds because (1) if $z \geq c_1$, then (25) directly follows from (23); (2) if $1 \leq z \leq c_1$, then the right-hand side (R.H.S.) of (25) is at least one (by assumption, $\beta > 1$) and the inequality trivially holds. The concentration inequality, that is, Equation (25), is now almost the same as the desired form in Theorem 3. The only difference is that it only holds for $n \geq N_p'$. This is not hard to resolve. The easiest approach, which we show in the following, is to refine the constants to ensure that when $1 \leq n < N_p'$, Equation (25) is trivially true. To this end, we note that $|n\bar{X}_n - n\mu_*| \leq 2Rn$. For each $1 \leq n < N_p'$, there is a corresponding $\bar{z}(n)$ such that $n^{\frac{\alpha}{\xi(1-\eta)}}\bar{z}(n) = 2Rn$. That is,

$$\bar{z}(n) \triangleq 2Rn^{1-\frac{\alpha}{\xi(1-\eta)}}, \quad 1 \leq n < N_p'.$$

This then implies that for each $1 \leq n < N_p'$, the following inequality trivially holds:

$$\mathbb{P}\left(n\bar{X}_n - n\mu_* \geq n^{\frac{\alpha}{\xi(1-\eta)}}z\right) \leq \frac{\bar{z}(n)^{\alpha-1}}{z^{\alpha-1}}, \quad \forall z \geq 1.$$

To see why, note that for each $1 \leq n < N_p'$: (1) if $z \geq \bar{z}(n)$, then $n^{\frac{\alpha}{\xi(1-\eta)}}z \geq 2Rn$ and the previous probability should be zero. Hence, any positive number on the R.H.S. makes the inequality trivially true; (2) if $1 \leq z < \bar{z}(n)$, the R.H.S. is at least one, which again makes the inequality hold. For convenience, define

$$c_2 \triangleq \max_{1 \leq n < N_p'} \bar{z}(n) = 2R(N_p'-1)^{1-\frac{\alpha}{\xi(1-\eta)}}. \tag{26}$$

Then, it is easy to see that for every $n \geq 1$ and every $z \geq 1$, we have

$$\mathbb{P}(n\bar{X}_n - n\mu_* \geq n^{\eta'}z) \leq \frac{\beta'}{z^{\xi'}},$$

where the constants are given by

$$\eta' = \frac{\alpha}{\xi(1-\eta)}, \tag{27}$$
$$\xi' = \alpha - 1, \tag{28}$$
$$\beta' = \max\left\{c_2, 2c_1^{\alpha-1}\cdot\max\left(\beta, \frac{2(K-1)}{(\alpha-1)(1+A(N_p'))^{\alpha-1}}\right)\right\}. \tag{29}$$

Finally, notice that because $\alpha \geq \xi\eta(1-\eta)$ and $\alpha < \xi(1-\eta)$, we have $1/2 \leq \eta \leq \eta' < 1$. Per (24), c_1 depends on $R, K, \Delta_{\min}, \beta, \xi$ and η. In addition, c_2 depends on $R, K, \Delta_{\min}, \beta, \xi, \alpha, \eta$ and N_p' depends on $R, K, \Delta_{\min}, \beta, \xi, \alpha, \eta$. Therefore, β' depends on $R, K, \Delta_{\min}, \beta, \xi, \alpha, \eta$. The other direction follows exactly the same reasoning, and this completes the proof of Theorem 3.

6.1. Proof of Lemma 1

By the choice of $A_i(t)$, s, and t, we have $B_{t,s} = \frac{\Phi(s,t^{-\alpha})}{s} \leq \frac{\Phi(A_i(t),t^{-\alpha})}{A_i(t)} \leq \frac{\Delta_i}{2}$. Therefore,

$$\begin{aligned}
\mathbb{P}(U_{i,s,t} > \mu_*) &= \mathbb{P}(\bar{X}_{i,s} + B_{t,s} > \mu_*) \\
&= \mathbb{P}(\bar{X}_{i,s} - \mu_i > \Delta_i - B_{t,s}) \\
&\leq \mathbb{P}(\bar{X}_{i,s} - \mu_i > B_{t,s}) \qquad \Delta_i \geq 2B_{t,s} \\
&\leq t^{-\alpha}. \qquad \text{by concentration (14).}
\end{aligned}$$

6.2. Proof of Lemma 2

If a suboptimal arm i is chosen at time $t+1$, that is, $I_{t+1}=i$, then at least one of the following two equations must be true: with notation $T_*(\cdot)=T_{i_*}(\cdot)$,

$$U_{i_*,T_*(t),t}\le\mu_*, \tag{30}$$

$$U_{i,T_i(t),t}>\mu_*. \tag{31}$$

Indeed, if both inequalities are false, we have $U_{i_*,T_*(t),t}>\mu_*\ge U_{i,T_i(t),t}$, which is a contradiction to $I_{t+1}=i$. We now use this fact to prove Lemma 2.

Case 1: $n>A_i(n)$. Such n exists because $A_i(n)$ grows with a polynomial order $O(n^{\frac{\alpha}{\xi(1-\eta)}})$ and $\alpha<\xi(1-\eta)$, that is, $A_i(n)=o(n)$. Then,

$$\begin{aligned}T_i(n)&=\sum_{t=0}^{n-1}\mathbb{I}\{I_{t+1}=i\}\overset{(a)}{=}1+\sum_{t=K}^{n-1}\mathbb{I}\{I_{t+1}=i\}\\&=1+\sum_{t=K}^{n-1}(\mathbb{I}\{I_{t+1}=i,T_i(t)<A_i(n)\}+\mathbb{I}\{I_{t+1}=i,T_i(t)\ge A_i(n)\})\\&\le A_i(n)+\sum_{t=K}^{n-1}\mathbb{I}\{I_{t+1}=i,T_i(t)\ge A_i(n)\},\end{aligned}$$

where equality (a) follows from the fact that $B_{t,s}=\infty$ if $s=0$.

To analyze the previous summation, we note that from (30) and (31),

$$\begin{aligned}\mathbb{I}\{I_{t+1}=i,T_i(t)\ge A_i(n)\}&\le\mathbb{I}\{U_{i_*,T_*(t),t}\le\mu_*\text{ or }U_{i,T_i(t),t}>\mu_*,T_i(t)\ge A_i(n)\}\\&\le\mathbb{I}\{U_{i,T_i(t),t}>\mu_*,T_i(t)\ge A_i(n)\}+\mathbb{I}\{U_{i_*,T_*(t),t}\le\mu_*,T_i(t)\ge A_i(n)\}\\&\le\mathbb{I}\{U_{i,T_i(t),t}>\mu_*,T_i(t)\ge A_i(n)\}+\mathbb{I}\{U_{i_*,T_*(t),t}\le\mu_*\}\\&=\mathbb{I}\{\exists s:A_i(n)\le s\le t,\text{s.t. }U_{i,s,t}>\mu_*\}\\&\quad+\mathbb{I}\{\exists s_*:1\le s_*\le t,\text{s.t. }U_{i_*,s_*,t}\le\mu_*\}.\end{aligned}$$

To summarize, we have proved that

$$\begin{aligned}\mathbb{E}[T_i(n)]&\le A_i(n)+\sum_{t=A_i(n)}^{n-1}\mathbb{P}((30)\text{ or }(31)\text{ is true, and }T_i(t)\ge A_i(n))\\&\le A_i(n)+\sum_{t=A_i(n)}^{n-1}\Bigg[\mathbb{P}(\underbrace{\exists s:A_i(n)\le s\le t,\text{s.t. }U_{i,s,t}>\mu_*}_{E_1})\\&\quad+\mathbb{P}(\underbrace{\exists s_*:1\le s_*\le t,\text{s.t. }U_{i_*,s_*,t}\le\mu_*}_{E_2})\Bigg].\end{aligned} \tag{32}$$

To complete the proof of Lemma 2, it suffices to bound the probabilities of the two events E_1 and E_2. To this end, we use a union bound:

$$\mathbb{P}(E_1)\le\sum_{s=A_i(n)}^{t}\mathbb{P}(U_{i,s,t}>\mu_*)\overset{(a)}{\le}\sum_{s=A_i(n)}^{t}t^{-\alpha}\le t\cdot t^{-\alpha}=t^{1-\alpha},$$

where the step (a) follows from $A_i(n)\ge A_i(t)$ and Lemma 1. We bound $\mathbb{P}(E_2)$ in a similar way:

$$\mathbb{P}(E_2)\le\sum_{s_*=1}^{t}\mathbb{P}(U_{i_*,s_*,t}\le\mu_*)=\sum_{s_*=1}^{t}\mathbb{P}(\bar{X}_{i_*,s_*}+B_{t,s_*}\le\mu_*)\overset{(a)}{\le}\sum_{s_*=1}^{t}t^{-\alpha}\le t^{1-\alpha},$$

where step (a) follows from concentration (cf. (14)). By substituting the bounds of $\mathbb{P}(E_1)$ and $\mathbb{P}(E_2)$ into (32), we have

$$\begin{aligned}\mathbb{E}[T_i(n)]&\le A_i(n)+\sum_{t=A_i(n)}^{n-1}2t^{1-\alpha}\\&\le A_i(n)+\int_{A_i(n)-1}^{\infty}2t^{1-\alpha}dt\quad\alpha>2\\&=A_i(n)+\frac{2(A_i(n)-1)^{2-\alpha}}{\alpha-2}\\&\le A_i(n)+\frac{2}{\alpha-2}\\&\le\left(\tfrac{2}{\Delta_i}\cdot\beta^{1/\xi}\right)^{\frac{1}{1-\eta}}\cdot n^{\frac{\alpha}{\xi(1-\eta)}}+\frac{2}{\alpha-2}+1.\end{aligned}$$

Case 2: $n\le A_i(n)$. If n is such that $n\le A_i(n)$, then the previous bound trivially holds because $T_i(n)\le n\le A_i(n)$. This completes the proof of Lemma 2.

6.3. Proof of Lemma 3

We first prove one direction, namely, $\mathbb{P}(n\mu_*-n\bar{X}_n\ge r_0x)$. The other direction follows the similar steps, and we will comment on that at the end of this proof. The general idea underlying the proof is to rewrite the quantity $n\mu_*-n\bar{X}_n$ as sums of terms that can be bounded using previous lemmas or assumptions. To begin with, note that

$$\begin{aligned}n\mu_*-n\bar{X}_n&=n\mu_*-\sum_{i=1}^{K}T_i(n)\bar{X}_{i,T_i(n)}\\&=n\mu_*-\sum_{t=1}^{T_*(n)}X_{i_*,t}-\sum_{i\ne i_*}T_i(n)\bar{X}_{i,T_i(n)}\\&=n\mu_*-\sum_{t=1}^{n}X_{i_*,t}+\sum_{t=T_*(n)+1}^{n}X_{i_*,t}-\sum_{i\ne i_*}\sum_{t=1}^{T_i(n)}X_{i,t}\\&\le n\mu_*-\sum_{t=1}^{n}X_{i_*,t}+2R\sum_{i\ne i_*}T_i(n),\end{aligned}$$

because $X_{i,t}\in[-R,R]$ for all i,t. Therefore, we have

$$\begin{aligned}\mathbb{P}(n\mu_*-n\bar{X}_n\ge r_0x)&\le\mathbb{P}\left(n\mu_*-\sum_{t=1}^{n}X_{i_*,t}+2R\sum_{i\ne i_*}T_i(n)\ge r_0x\right)\\&\le\mathbb{P}\left(n\mu_*-\sum_{t=1}^{n}X_{i_*,t}\ge n^{\eta}x\right)+\sum_{i\ne i_*}\mathbb{P}(T_i(n)\ge(3+A(n))x),\end{aligned} \tag{33}$$

where the last inequality follows from the union bound.

To prove the theorem, we now bound the two terms in (33). By our concentration assumption, we can upper bound the first term as follows:

$$\mathbb{P}\left(n\mu_*-\sum_{t=1}^{n}X_{i_*,t}\ge n^{\eta}x\right)\le\frac{\beta}{x^{\xi}}. \tag{34}$$

Next, we bound each term in the summation of (33). Fix n and a suboptimal edge i. Let u be an integer satisfying $u\ge A(n)$. For any $\tau\in\mathbb{R}$, consider the following two events:

$$E_1=\{\text{For each integer }t\in[u,n],\text{ we have }U_{i,u,t}\le\tau\},$$

$E_2 = \{\text{For each integer } s \in [1, n-u], \text{ we have } U_{i_*,s,u+s} > \tau\}$.

As a first step, we want to show that

$$E_1 \cap E_2 \Rightarrow T_i(n) \le u. \tag{35}$$

To this end, let us condition on both events E_1 and E_2. Recall that $B_{t,s}$ is nondecreasing with respect to t. Then, for each s such that $1 \le s \le n-u$, and each t such that $u+s \le t \le n$, it holds that

$$U_{i_*,s,t} = \bar{X}_{i_*,s} + B_{t,s} \ge \bar{X}_{i_*,s} + B_{u+s,s} = U_{i_*,s,u+s} > \tau \ge U_{i,u,t}.$$

This implies that $T_i(n) \le u$. To see why, suppose that $T_i(n) > u$ and denote by t' the first time that arm i has been played u times, that is, $t' = \min\{t : t \le n, T_i(t) = u\}$. By definition, $t' \ge u + T_*(t')$. Hence, for any time t such that $t' < t \le n$, the previous inequality implies that $U_{i_*,T_*(t),t} > U_{i,u,t}$. That is, i^* always has a higher upper confidence bound than i, and arm i will not be selected; that is, arm i will not be played the $(u+1)$ th time. This contradicts our assumption that $T_i(n) > u$, and hence we have the inequality $T_i(n) \le u$.

To summarize, we have established the fact that $E_1 \cap E_2 \Rightarrow T_i(n) \le u$. As a result, we have

$$\begin{aligned}\{T_i(n) > u\} &\subset (E_1^c \cup E_2^c)\\ &= (\{\exists t : u \le t \le n \,\text{s.t.}\, U_{i,u,t} > \tau\}\\ &\quad \cup \{\exists\, \text{s} : 1 \le s \le n-u, \text{s.t. } U_{i_*,s,u+s} \le \tau\}).\end{aligned}$$

Using union bound, we obtain that

$$\mathbb{P}(T_i(n) > u) \le \sum_{t=u}^{n} \mathbb{P}(U_{i,u,t} > \tau) + \sum_{s=1}^{n-u} \mathbb{P}(U_{i_*,s,u+s} \le \tau). \tag{36}$$

For the previous bound, we are free to choose u and τ as long as $u \ge A(n)$. To connect with our goal (cf. (33)), in the following, we set $u = \lfloor (1+A(n))x \rfloor + 1$ (recall that $x \ge 1$) and $\tau = \mu_*$ to bound $\mathbb{P}(T_i(n) > u)$. Because $u \ge A(n) \ge A_i(n)$, by Lemma 1, we have

$$\begin{aligned}\sum_{t=u}^{n} \mathbb{P}(U_{i,u,t} > \mu_*) &\le \sum_{t=u}^{n} t^{-\alpha} \le \int_{u-1}^{\infty} t^{-\alpha} dt = \frac{(u-1)^{1-\alpha}}{\alpha-1}\\ &= \frac{(\lfloor (1+A(n))x \rfloor)^{1-\alpha}}{\alpha-1} \le \frac{((1+A(n))x)^{1-\alpha}}{\alpha-1}.\end{aligned}$$

As for the second summation in the R.H.S. of (36), we have that

$$\begin{aligned}\sum_{s=1}^{n-u} \mathbb{P}(U_{i_*,s,u+s} \le \tau) &= \sum_{s=1}^{n-u} \mathbb{P}(U_{i_*,s,u+s} \le \mu_*)\\ &= \sum_{s=1}^{n-u} \mathbb{P}(\bar{X}_{i_*,s} + B_{u+s,s} \le \mu_*)\\ &\le \sum_{s=1}^{n-u} (s+u)^{-\alpha} = \sum_{t=1+u}^{n} t^{-\alpha}\\ &\le \int_{u-1}^{\infty} t^{-\alpha} dt = \frac{(u-1)^{1-\alpha}}{\alpha-1} \le \frac{((1+A(n))x)^{1-\alpha}}{\alpha-1},\end{aligned}$$

where the first inequality follows from the concentration property (cf. (14)). Combining the previous inequalities and note that $(3+A(n))x > \lfloor (1+A(n))x \rfloor + 1$:

$$\begin{aligned}\mathbb{P}(T_i(n) \ge (3+A(n))x) &\le \mathbb{P}(T_i(n) > u)\\ &\le \frac{2((1+A(n))x)^{1-\alpha}}{\alpha-1}.\end{aligned} \tag{37}$$

Substituting (34) and (37) into (33), we obtain

$$\mathbb{P}(n\mu_* - n\bar{X}_n \ge r_0 x) \le \frac{\beta}{x^{\xi}} + \sum_{i \ne i_*} \frac{2((1+A(n))x)^{1-\alpha}}{\alpha-1},$$

which is the desired inequality in Lemma 3.

To complete the proof, we need to consider the other direction, that is, $\mathbb{P}(n\bar{X}_n - n\mu_* \ge r_0 x)$. The proof is almost identical. Note that

$$\begin{aligned}n\bar{X}_n - n\mu_* &= \sum_{i=1}^{K} T_i(n)\bar{X}_{i,T_i(n)} - n\mu_*\\ &= \sum_{t=1}^{n} X_{i_*,t} - n\mu_* - \sum_{t=T_*(n)+1}^{n} X_{i_*,t} + \sum_{i \ne i_*} \sum_{t=1}^{T_i(n)} X_{i,t}\\ &\le \sum_{t=1}^{n} X_{i_*,t} - n\mu_* + 2R \sum_{i \ne i_*} T_i(n),\end{aligned}$$

because $X_{i,t} \in [-R, R]$ for all i, t. Therefore,

$$\begin{aligned}\mathbb{P}(n\bar{X}_n - n\mu_* \ge r_0 x) &\le \mathbb{P}\left(\sum_{t=1}^{n} X_{i_*,t} - n\mu_* + 2R \sum_{i \ne i_*} T_i(n) \ge r_0 x\right)\\ &\le \mathbb{P}\left(\sum_{t=1}^{n} X_{i_*,t} - n\mu_* \ge n^{\eta} x\right)\\ &\quad + \sum_{i \ne i_*} \mathbb{P}(T_i(n) \ge (3+A_i(n))x).\end{aligned}$$

The desired inequality then follows exactly from the same reasoning of our previous proof.

7. Analysis of MCTS and Proof of Theorem 1

In this section, we give a complete analysis for the fixed-depth MCTS algorithm illustrated in Algorithm 1 and prove Theorem 1. In effect, as discussed in Section 3, one can view a depth-H MCTS as a simulated version of H steps value function iteration. Given the current value function proxy $\hat{V}$, let $V^{(H)}(\cdot)$ be the value function estimation after H steps of value function iteration starting with the proxy $\hat{V}$. Then, we prove the result in two parts. First, we argue that because of the MCTS sampling process, the mean of the empirical estimation of value function at the query node s, or the root node of MCTS tree, is within $O(n^{\eta-1})$ of $V^{(H)}(s)$ after n simulations, with the given proxy $\hat{V}$ being the input to the MCTS algorithm. Second, we argue that $V^{(H)}(s)$ is within $\gamma^H \|\hat{V} - V^*\|_\infty \le \gamma^H \varepsilon_0$ of the optimal value function. Putting this together leads to Theorem 1.

We start by a preliminary probabilistic lemma in Section 7.1 that will be useful throughout. Sections 7.2 and 7.3 argue the first part of the proof as explained previously. Section 7.4 provides proof of the second part. Section 7.5 concludes the proof of Theorem 1.

7.1. Preliminary

We state the following probabilistic lemma that is useful throughout. Proof can be found in Section 7.6.

Lemma 4. *Consider real-valued random variables X_i, Y_i for $i \geq 1$ such that Xs are independent and identically distributed taking values in $[-B, B]$ for some $B > 0$, Xs are independent of Ys, and Ys satisfy*

A. *Convergence: for $n \geq 1$, with notation $\bar{Y}_n = \frac{1}{n}\left(\sum_{i=1}^{n} Y_i\right)$,*

$$\lim_{n\to\infty} \mathbb{E}\left[\bar{Y}_n\right] = \mu_Y.$$

B. *Concentration: there exist constants, $\beta > 1$, $\xi > 0$, $1/2 \leq \eta < 1$ such that for $n \geq 1$ and $z \geq 1$,*

$$\mathbb{P}(n\bar{Y}_n - n\mu_Y \geq n^{\eta} z) \leq \frac{\beta}{z^{\xi}}, \quad \mathbb{P}(n\bar{Y}_n - n\mu_Y \leq -n^{\eta} z) \leq \frac{\beta}{z^{\xi}}.$$

Let $Z_i = X_i + \rho Y_i$ for some $\rho > 0$. Then, Zs satisfy

A. *Convergence: for $n \geq 1$, with notation $\bar{Z}_n = \frac{1}{n}\left(\sum_{i=1}^{n} Z_i\right)$, and $\mu_X = \mathbb{E}[X_1]$,*

$$\lim_{n\to\infty} \mathbb{E}\left[\bar{Z}_n\right] = \mu_X + \rho\mu_Y.$$

B. *Concentration: there exist constant $\beta' > 1$ depending upon ρ, ξ, β and B, such that for $n \geq 1$ and $z \geq 1$,*

$$\mathbb{P}(n\bar{Z}_n - n(\mu_X + \rho\mu_Y) \geq n^{\eta} z) \leq \frac{\beta'}{z^{\xi}},$$

$$\mathbb{P}(n\bar{Z}_n - n(\mu_X + \rho\mu_Y) \leq -n^{\eta} z) \leq \frac{\beta'}{z^{\xi}}.$$

7.2. Analyzing Leaf Level *H*

The goal is to understand the empirical reward observed at the query node for MCTS or the root node of the MCTS tree. In particular, we argue that the mean of the empirical reward at the root node of the MCTS tree is within $O(n^{\eta-1})$ of the mean reward obtained at it assuming access to infinitely many samples. We start by analyzing the reward collected at the nodes that are at leaf level H and level $H-1$.

The nodes at leaf level, that is, level H, are children of nodes at level $H-1$ in the MCTS tree. Let there be n_{H-1} nodes at level $H-1$ corresponding to states $s_{1,H-1}, \ldots, s_{n_{H-1},H-1} \in \mathcal{S}$. Consider node $i \in [n_{H-1}]$ at level $H-1$, corresponding to state $s_{i,H-1}$. As part of the algorithm, whenever this node is visited, one of the K feasible actions is taken. When an action $a \in [K]$ is taken, the node $s'_H = s_{i,H-1} \circ a$, at the leaf level H is reached. This results in reward at node $s_{i,H-1}$ (at level $H-1$) being equal to $\mathcal{R}(s_{i,H-1}, a) + \gamma\tilde{v}^{(H)}(s'_H)$. Here, for each $s \in \mathcal{S}$ and $a \in [K]$, the reward $\mathcal{R}(s,a)$ is an independent, bounded random variable taking value in $[-R_{\max}, R_{\max}]$ with distribution dependent on s, a; $\tilde{v}^{(H)}(\cdot)$ is the input of value function proxy to the MCTS algorithm denoted as $\hat{V}(\cdot)$, and $\gamma \in [0,1)$ is the discount factor. Recall that $\varepsilon_0 = \|\hat{V} - V^*\|_\infty$ and $\|V^*\|_\infty \leq V_{\max}$. Therefore, $\|\tilde{v}^{(H)}\|_\infty = \|\hat{V}\|_\infty \leq V_{\max} + \varepsilon_0$, and the reward collected at node $s_{i,H-1}$ by following any action is bounded, in absolute value, by $\tilde{R}_{\max^{(H-1)}} = R_{\max} + \gamma(V_{\max} + \varepsilon_0)$.

As part of the MCTS algorithm as described in (5), when node $s_{i,H-1}$ is visited for the $t+1$ time with $t \geq 0$, the action taken is

$$\arg\max_{a\in\mathcal{A}} \left\{ \frac{1}{u_a} \sum_{j=1}^{u_a} (r(s_{i,H-1}, a)(j) + \gamma\tilde{v}^{(H)}(s_{i,H-1} \circ a)(j)) + \frac{(\beta^{(H)})^{1/\xi^{(H)}} \cdot (t)^{\alpha^{(H)}/\xi^{(H)}}}{(u_a)^{1-\eta^{(H)}}} \right\},$$

where $u_a \leq t$ is the number of times action a has been chosen thus far at state $s_{i,H-1}$ in the t visits thus far, $r(s_{i,H-1}, a)(j)$ is the jth sample of random variable per distribution $\mathcal{R}(s_{i,H-1}, a)$, and $\tilde{v}^{(H)}(s_{i,H-1} \circ a)(j)$ is the reward evaluated at leaf node $s_{i,H-1} \circ a$ for the jth time. For all j, the reward evaluated at leaf node $s_{i,H-1} \circ a$ is the same and equals to $\tilde{v}^{(H)}(\cdot)$, the input value function proxy for the algorithm. When $u_a = 0$, we use notation ∞ to represent quantity inside the arg max. The net discounted reward collected by node $s_{i,H-1}$ during its total of $t \geq 1$ visits is simply the sum of rewards obtained by selecting the actions per the policy, which includes the reward associated with taking an action and the evaluation of $\tilde{v}^{(H)}(\cdot)$ for appropriate leaf node, discounted by γ. In effect, at each node $s_{i,H-1}$, we are using the UCB policy described in Section 5 with parameters $\alpha^{(H)}, \beta^{(H)}, \xi^{(H)}, \eta^{(H)}$ with K possible actions, where the rewards collected by playing any of these K actions each time is simply the summation of bounded independent and identical (for a given action) random variable and a deterministic evaluation. By applying Lemma 4, where Xs correspond to independent rewards, $\rho = \gamma$, and Ys correspond to deterministic evaluations of $\tilde{v}^{(H)}(\cdot)$, we obtain that for given $\xi^{(H)} > 0$ and $\eta^{(H)} \in \left[\frac{1}{2}, 1\right)$, there exists $\beta^{(H)}$ such that the collected rewards at $s_{i,H-1}$ (i.e., sum of i.i.d. reward and deterministic evaluations) satisfy the convergence property (cf. (13)) and concentration property (cf. (14)) stated in Section 5. Therefore, by an application of Theorem 3, we conclude Lemma 5. We define some notations first:

$$\begin{aligned}
&\mu_a^{(H-1)}(s_{i,H-1}) = \mathbb{E}\left[\mathcal{R}(s_{i,H-1}, a)\right] + \gamma\tilde{v}^{(H)}(s_{i,H-1} \circ a),\\
&\mu_*^{(H-1)}(s_{i,H-1}) = \max_{a\in[K]} \mu_a^{(H-1)}(s_{i,H-1})\\
&a_*^{(H-1)}(s_{i,H-1}) \in \arg\max_{a\in[K]} \mu_a^{(H-1)}(s_{i,H-1})\\
&\Delta_{\min}^{(H-1)}(s_{i,H-1}) = \mu_*^{(H-1)}(s_{i,H-1}) - \max_{a\neq a_*^{(H-1)}(s_{i,H-1})} \mu_a^{(H-1)}(s_{i,H-1}).
\end{aligned} \tag{38}$$

We shall assume that the maximizer in the set $\arg\max_{a\in[K]}\mu_a^{(H-1)}(s_{i,H-1})$ is unique, that is, $\Delta_{\min^{(H-1)}}(s_{i,H-1})>0$. Further note that all rewards belong to $\left[-\tilde{R}_{\max^{(H-1)}},\tilde{R}_{\max^{(H-1)}}\right]$.

Lemma 5. *Consider a node corresponding to state $s_{i,H-1}$ at level $H-1$ within the MCTS for $i\in[n_{H-1}]$. Let $\tilde{v}^{(H-1)}(s_{i,H-1})_n$ be the total discounted reward collected at $s_{i,H-1}$ during $n\geq 1$ visits of it, to one of its K leaf nodes under the UCB policy. Then, for the choice of appropriately large $\beta^{(H)}>0$, for a given $\xi^{(H)}>0$, $\eta^{(H)}\in\left[\frac{1}{2},1\right)$ and $\alpha^{(H)}>2$, we have*

A. *Convergence:*

$$\left|\mathbb{E}\left[\frac{1}{n}\tilde{v}^{(H-1)}(s_{i,H-1})_n\right]-\mu_*^{(H-1)}(s_{i,H-1})\right| \leq \frac{2\tilde{R}_{\max^{(H-1)}}(K-1)\cdot\left(\left(\frac{2(\beta^{(H)})^{\frac{1}{\xi^{(H)}}}}{\Delta_{\min}^{(H-1)}(s_{i,H-1})}\right)^{\frac{1}{1-\eta^{(H)}}}\cdot n^{\frac{\alpha^{(H)}}{\xi^{(H)}(1-\eta^{(H)})}}+\frac{2}{\alpha^{(H)}-2}+1\right)}{n}.$$

B. *Concentration: there exist constants, $\beta'>1$ and $\xi'>0$ and $1/2\leq\eta'<1$ such that for every $n\geq 1$ and every $z\geq 1$,*

$$\mathbb{P}(\tilde{v}^{(H-1)}(s_{i,H-1})_n-n\mu_*^{(H-1)}(s_{i,H-1})\geq n^{\eta'}z)\leq\frac{\beta'}{z^{\xi'}},$$
$$\mathbb{P}(\tilde{v}^{(H-1)}(s_{i,H-1})_n-n\mu_*^{(H-1)}(s_{i,H-1})\leq -n^{\eta'}z)\leq\frac{\beta'}{z^{\xi'}},$$

where $\eta'=\frac{\alpha^{(H)}}{\xi^{(H)}(1-\eta^{(H)})}$, $\xi'=\alpha^{(H)}-1$, and β' is a large enough constant that is function of parameters $\alpha^{(H)},\beta^{(H)},\xi^{(H)},\eta^{(H)},\tilde{R}_{\max^{(H-1)}},K,\Delta_{\min}^{(H-1)}(s_{i,H-1})$.

Let $\Delta_{\min^{(H-1)}}=\min_{i\in[n_{H-1}]}\Delta_{\min}^{(H-1)}(s_{i,H-1})$. Then, the rate of convergence for each node $s_{i,H-1}$, $i\in[n_{H-1}]$ can be uniformly simplified as

$$\begin{aligned}\delta_n^{(H-1)}&=\frac{2\tilde{R}_{\max^{(H-1)}}(K-1)\cdot\left(\left(\frac{2(\beta^{(H)})^{\frac{1}{\xi^{(H)}}}}{\Delta_{\min^{(H-1)}}}\right)^{\frac{1}{1-\eta^{(H)}}}\cdot n^{\frac{\alpha^{(H)}}{\xi^{(H)}(1-\eta^{(H)})}}+\frac{2}{\alpha^{(H)}-2}+1\right)}{n}\\&=\Theta\left(n^{\frac{\alpha^{(H)}}{\xi^{(H)}(1-\eta^{(H)})}-1}\right)\\&\overset{(a)}{=}O(n^{\eta-1}),\end{aligned}$$

where (a) holds because $\alpha^{(H)}=\xi^{(H)}(1-\eta^{(H)})\eta^{(H)}$, $\eta^{(H)}=\eta$. It is worth remarking that $\mu_*^{(H-1)}(s_{i,H-1})$, as defined in (38), is precisely the value function estimation for $s_{i,H-1}$ at the end of one step of value iteration starting with $\hat{V}$.

7.3. Recursion: Going from Level h to $h-1$

Lemma 5 suggests that the necessary assumption of Theorem 3, that is, (13) and (14), is satisfied by $\tilde{v}_n^{(H-1)}$ for each node or state at level $H-1$, with $\alpha^{(H-1)}$, $\xi^{(H-1)},\eta^{(H-1)}$ as defined per relationship (6)–(8) and with appropriately defined large enough constant $\beta^{(H-1)}$. We shall argue that result similar to Lemma 5, but for node at level $H-2$, continues to hold with parameters $\alpha^{(H-2)},\xi^{(H-2)},\eta^{(H-2)}$ as defined per relationship (6)–(8) and with appropriately defined large enough constant $\beta^{(H-2)}$. A similar argument will continue to apply going from level h to $h-1$ for all $h\leq H-1$. That is, we shall assume that the necessary assumption of Theorem 3, that is, (13) and (14), holds for $\tilde{v}^{(h)}(\cdot)$, for all nodes at level h with $\alpha^{(h)},\xi^{(h)},\eta^{(h)}$ as defined per relationship (6)–(8) and with appropriately defined large enough constant $\beta^{(h)}$, and then argue that such holds for nodes at level $h-1$ as well. This will, using mathematical induction, allow us to prove the results for all $h\geq 1$.

To that end, consider any node at level $h-1$. Let there be n_{h-1} nodes at level $h-1$ corresponding to states $s_{1,h-1},\ldots,s_{n_{h-1},h-1}\in\mathcal{S}$. Consider a node corresponding to state $s_{i,h-1}$ at level $h-1$ within the MCTS for $i\in[n_{h-1}]$. As part of the algorithm, whenever this node is visited, one of the K feasible action is taken. When an action $a\in[K]$ is taken, the node $s_h'=s_{i,h-1}\circ a$, at the level h is reached. This results in reward at node $s_{i,h-1}$ at level $h-1$ being equal to $\mathcal{R}(s_{i,h-1},a)+\gamma\tilde{v}^{(h)}(s_h')$. As noted before, $\mathcal{R}(s,a)$ is an independent, bounded valued random variable while $\tilde{v}^{(h)}(\cdot)$ is effectively collected by following a path all the way to the leaf level. Inductively, we assume that $\tilde{v}^{(h)}(\cdot)$ satisfies the convergence and concentration property for each node or state at level h, with $\alpha^{(h)},\xi^{(h)},\eta^{(h)}$ as defined per relationship (6)–(8) and with appropriately defined large enough constant $\beta^{(h)}$. Therefore, by an application of Lemma 4, it follows that this combined reward continues to satisfy (13) and (14), with $\alpha^{(h)},\xi^{(h)},\eta^{(h)}$ as defined per relationship (6)–(8) and with a large enough constant that we shall denote as $\beta^{(h)}$. These constants are used by the MCTS policy. By an application of Theorem 3, we can obtain the following Lemma 6 regarding the convergence and concentration properties for the reward sequence collected at node $s_{i,h-1}$ at level $h-1$. Similar to the notation in Equation (38), let

$$\begin{aligned}\mu_a^{(h-1)}(s_{i,h-1})&=\mathbb{E}\left[\mathcal{R}(s_{i,h-1},a)\right]+\gamma\mu_*^{(h)}(s_{i,h-1}\circ a)\\\mu_*^{(h-1)}(s_{i,h-1})&=\max_{a\in[K]}\mu_a^{(h-1)}(s_{i,h-1})\\a_*^{(h-1)}(s_{i,h-1})&\in\arg\max_{a\in[K]}\mu_a^{(h-1)}(s_{i,h-1})\\\Delta_{\min}^{(h-1)}(s_{i,h-1})&=\mu_*^{(h-1)}(s_{i,h-1})-\max_{a\neq a_*^{(h-1)}(s_{i,h-1})}\mu_a^{(h-1)}(s_{i,h-1}).\end{aligned}\tag{39}$$

Again, we shall assume that the maximizer in the set $\arg\max_{a\in[K]}\mu_a^{(h-1)}(s_{i,h-1})$ is unique, that is, $\Delta_{\min}^{(h-1)}(s_{i,h-1})>0$. Define $\tilde{R}_{\max^{(h-1)}}=R_{\max}+\gamma\tilde{R}_{\max^{(h)}}$, where $\tilde{R}^{(H)}=V_{\max}+\varepsilon_0$. All rewards collected at level $h-1$ belong to $\left[-\tilde{R}_{\max}^{(h-1)},\tilde{R}_{\max}^{(h-1)}\right]$.

Lemma 6. *Consider a node corresponding to state $s_{i,h-1}$ at level $h-1$ within the MCTS for $i\in[n_{h-1}]$. Let $\tilde{v}^{(h-1)}(s_{i,h-1})_n$ be the total discounted reward collected at $s_{i,h-1}$ during $n\geq 1$ visits. Then, for the choice of appropriately*

large $\beta^{(h)} > 0$, for a given $\xi^{(h)} > 0$, $\eta^{(h)} \in [\frac{1}{2}, 1)$ and $\alpha^{(h)} > 2$, we have

A. *Convergence:*

$$\left|\mathbb{E}\left[\frac{1}{n}\tilde{v}^{(h-1)}(s_{i,h-1})_n\right] - \mu_*^{(h-1)}(s_{i,h-1})\right| \le \frac{2\tilde{R}_{\max^{(h-1)}}(K-1)\cdot\left(\left(\frac{2(\beta^{(h)})^{\frac{1}{\xi^{(h)}}}}{\Delta_{\min}^{(h-1)}(s_{i,h-1})}\right)^{\frac{1}{1-\eta^{(h)}}}\cdot n^{\frac{\alpha^{(h)}}{\xi^{(h)}(1-\eta^{(h)})}} + \frac{2}{\alpha^{(h)}-2} + 1\right)}{n}.$$

B. *Concentration: there exist constants, $\beta' > 1$ and $\xi' > 0$ and $1/2 \le \eta' < 1$ such that for $n \ge 1, z \ge 1$,*

$$\mathbb{P}(\tilde{v}^{(h-1)}(s_{i,h-1})_n - n\mu_*^{(h-1)}(s_{i,h-1}) \ge n^{\eta'}z) \le \frac{\beta'}{z^{\xi'}},$$
$$\mathbb{P}(\tilde{v}^{(h-1)}(s_{i,h-1})_n - n\mu_*^{(h-1)}(s_{i,h-1}) \le -n^{\eta'}z) \le \frac{\beta'}{z^{\xi'}},$$

where $\eta' = \frac{\alpha^{(h)}}{\xi^{(h)}(1-\eta^{(h)})}$, $\xi' = \alpha^{(h)} - 1$, and β' is a large enough constant that is function of parameters $\alpha^{(h)}, \beta^{(h)}, \xi^{(h)}, \eta^{(h)}, \tilde{R}_{\max^{(h-1)}}, K, \Delta_{\min}^{(h-1)}(s_{i,h-1})$.

As before, let us define $\Delta_{\min}^{(h-1)} = \min_{i\in[n_{h-1}]}\Delta_{\min}^{(h-1)}(s_{i,h-1})$. Similarly, we can show that for every node $s_{i,h-1}$, $i \in [n_{h-1}]$, the rate of convergence in Lemma 6 can be uniformly simplified as

$$\delta_n^{(h-1)} = \frac{2\tilde{R}_{\max^{(h-1)}}(K-1)\cdot\left(\left(\frac{2(\beta^{(h)})^{\frac{1}{\xi^{(h)}}}}{\Delta_{\min^{(h-1)}}}\right)^{\frac{1}{1-\eta^{(h)}}}\cdot n^{\frac{\alpha^{(h)}}{\xi^{(h)}(1-\eta^{(h)})}} + \frac{2}{\alpha^{(h)}-2} + 1\right)}{n} = \Theta\left(n^{\frac{\alpha^{(h)}}{\xi^{(h)}(1-\eta^{(h)})}-1}\right) = O(n^{\eta-1}),$$

where the last equality holds as $\alpha^{(h)} = \xi^{(h)}(1-\eta^{(h)})\eta^{(h)}$ and $\eta^{(h)} = \eta$. Again, it is worth remarking, inductively, that $\mu_*^{(h-1)}(s_{i,h-1})$ is precisely the value function estimation for $s_{i,h-1}$ at the end of $H-h+1$ steps of value iteration starting with $\hat{V}$.

Remark 1 (Recursive Relation Among Parameters). With the previous development, we are ready to elaborate our choice of parameters in Theorem 1, defined recursively via Equations (6)–(8). In essence, those parameter requirements originate from our analysis of the nonstationary MAB, that is, Theorem 3. Recall that, from our previous analysis, the key to establish the MCTS guarantee is to recursively argue the convergence and the polynomial concentration properties at each level; that is, we recursively solve the nonstationary MAB problem at each level. To do so, we apply our result on the nonstationary MAB (Theorem 3) recursively at each level. Importantly, recall that Theorem 3 only holds when $\xi\eta(1-\eta) \le \alpha < \xi(1-\eta)$ and $\alpha > 2$, under which it leads to the recursive conclusions $\eta' = \frac{\alpha}{\xi(1-\eta)}$ and $\xi' = \alpha - 1$. Using our notation with superscript indicating the levels, this means that apart from the parameters at the leaf level (level H) that could be freely chosen, we must choose parameters of other levels recursively so that the following conditions hold:

$$\alpha^{(h)} > 2, \quad \xi^{(h)}\eta^{(h)}(1-\eta^{(h)}) \le \alpha^{(h)} < \xi^{(h)}(1-\eta^{(h)}),$$
$$\xi^{(h)} = \alpha^{(h+1)} - 1 \text{ and } \quad \eta^{(h)} = \frac{\alpha^{(h+1)}}{\xi^{(h+1)}(1-\eta^{(h+1)})}.$$

It is not hard to see that the conditions in Theorem 1 guarantee this. There might be other sequences of parameters satisfying the requirements, but our particular choice gives cleaner analysis as presented in this paper.

7.4. Error Analysis for Value Function Iteration

We now move to the second part of the proof. The value function iteration improves the estimation of optimal value function by iterating Bellman equation. In effect, the MCTS tree is "unrolling" H steps of such an iteration. Precisely, let $V^{(h)}(\cdot)$ denote the value function after h iterations starting with $V^{(0)} = \hat{V}$. By definition, for any $h \ge 0$ and $s \in \mathcal{S}$,

$$V^{(h+1)}(s) = \max_{a\in[K]}(\mathbb{E}[\mathcal{R}(s,a)] + \gamma V^{(h)}(s \circ a)). \tag{40}$$

Recall that value iteration is contractive with respect to $\|\cdot\|_\infty$ norm (Bertsekas 2017). That is, for any $h \ge 0$,

$$\|V^{(h+1)} - V^*\|_\infty \le \gamma\|V^{(h)} - V^*\|_\infty. \tag{41}$$

As remarked earlier, $\mu_*^{(h-1)}(s_{i,h-1})$, the mean reward collected at node $s_{i,h-1}$ for $i \in [n_{h-1}]$ for any $h \ge 1$, is precisely $V^{(H-h+1)}(s_{i,h-1})$ starting with $V^{(0)} = \hat{V}$, the input to MCTS policy. Therefore, the mean reward collected at root node $s^{(0)}$ of the MCTS tree satisfies $\mu_*^{(0)}(s^{(0)}) = V^{(H)}(s^{(0)})$. Using (41), we obtain the following lemma.

Lemma 7. *The mean reward collected under the MCTS policy at root note $s^{(0)}$, $\mu_*^{(0)}(s^{(0)})$, starting with input value function proxy $\hat{V}$ is such that*

$$\left|\mu_*^{(0)}(s^{(0)}) - V^*(s^{(0)})\right| \le \gamma^H\|\hat{V} - V^*\|_\infty. \tag{42}$$

7.5. Completing Proof of Theorem 1

In summary, using Lemma 6, we conclude that the recursive relationship going from level h to $h-1$ holds for all $h \ge 1$ with level 0 being the root. At root $s^{(0)}$, the query state that is input to the MCTS policy, we have that after n total simulations of MCTS, the empirical average of the rewards over these n trial, $\frac{1}{n}\tilde{v}^{(0)}(s_0)_n$ is such that (using the fact that $\alpha^{(0)} = \xi^{(0)}(1-\eta^{(0)})\eta^{(0)}$)

$$\left|\mathbb{E}\left[\frac{1}{n}\tilde{v}^{(0)}(s_0)_n\right] - \mu_*^{(0)}\right| = O\left(n^{\frac{\alpha^{(0)}}{\xi^{(0)}(1-\eta^{(0)})}-1}\right) = O(n^{\eta-1}), \tag{43}$$

where $\mu_*^{(0)}$ is the value function estimation for $s^{(0)}$ after H iterations of value function iteration starting with $\hat{V}$. By Lemma 7, we have

$$\left|\mu_*^{(0)} - V^*(s^{(0)})\right| \le \gamma^H\varepsilon_0, \tag{44}$$

because $\varepsilon_0 = \|\hat{V} - V^*\|_\infty$. Combining (43) and (44),

$$\left|\mathbb{E}\left[\frac{1}{n}\tilde{v}^{(0)}(s_0)_n\right] - V^*(s^{(0)})\right| \le \gamma^H \varepsilon_0 + O(n^{\eta-1}). \tag{45}$$

This concludes the proof of Theorem 1.

7.6. Proof of Lemma 4

The convergence property, $\lim_{n\to\infty}\mathbb{E}[\bar{Z}_n] = \mu_X + \rho\mu_Y$, follows simply by linearity of expectation. For concentration, consider the following: because Xs are i.i.d. bounded random variables taking value in $[-B, B]$, by Hoeffding's inequality (Hoeffding 1963), we have that for $t \ge 0$,

$$\begin{aligned}\mathbb{P}(n\bar{X}_n - n\mu_X \ge nt) &\le \exp\left(-\frac{t^2 n}{2B^2}\right),\\ \mathbb{P}(n\bar{X}_n - n\mu_X \le -nt) &\le \exp\left(-\frac{t^2 n}{2B^2}\right).\end{aligned} \tag{46}$$

Therefore,

$$\begin{aligned}\mathbb{P}(n\bar{Z}_n - n(\mu_X + \rho\mu_Y) \ge n^\eta z) &\le \mathbb{P}\left(n\bar{X}_n - n\mu_X \ge \frac{n^\eta z}{2}\right)\\ &\quad + \mathbb{P}\left(n\bar{Y}_n - n\mu_Y \ge \frac{n^\eta z}{2\rho}\right)\\ &\le \exp\left(-\frac{z^2 n^{2\eta-1}}{8B^2}\right) + \frac{\beta 2^\xi \rho^\xi}{z^\xi}\\ &\le \frac{\beta'}{z^\xi},\end{aligned} \tag{47}$$

where β' is a large enough constant depending on ρ, ξ, β, and B. The other side of the inequality follows similarly. This completes the proof.

8. Proof of Theorem 2

First, we establish a useful property of nearest neighbor supervised learning presented in Section 4.2. This is stated in Section 8.1. We will use it, along with the guarantees obtained for MCTS in Theorem 1 to establish Theorem 2 in Section 8.2. Throughout, we shall assume the setup of Theorem 2.

8.1. Guarantees for Supervised Learning

Let $\delta \in (0,1)$ be given. As stated in Section 4.2, let $K(\delta, d) = \Theta(\delta^{-d})$ be the collection of balls of radius δ, say c_i, $i \in [K(\delta,d)]$, so that they cover $\mathcal{S}$, that is, $\mathcal{S} \subset \bigcup_{i\in[K(\varepsilon,d)]} c_i$. Also, by construction, each of these balls have intersection with $\mathcal{S}$ whose volume is at least $C_d\delta^d$. Let $S = \{s_i : i \in [N]\}$ denote N state samples from $\mathcal{S}$ uniformly at random and independent of each other. For each state $s \in \mathcal{S}$, let $V : \mathcal{S} \to [-V_{\max}, V_{\max}]$ be such that $|\mathbb{E}[V(s)] - V^*(s)| \le \Delta$. Let the nearest neighbor supervised learning described in Section 4.2 produce estimate $\hat{V} : \mathcal{S} \to \mathbb{R}$ using labeled data points $(s_i, V(s_i))_{i\in[N]}$. Then, we claim the following guarantee. Proof can be found in Section 8.3.

Lemma 8. *Under the previously described setup, as long as* $N \ge 32\max\left(1, \delta^{-2}V_{\max^2}\right)C_d^{-1}\delta^{-d}\log\frac{K(\delta,d)}{\delta}$, *that is,* $N = \Omega\left(d\delta^{-d-2}\log\delta^{-1}\right)$,

$$\mathbb{E}\left[\sup_{s\in\mathcal{S}}|\hat{V}(s) - V^*(s)|\right] \le \Delta + (C+1)\delta + \frac{4V_{\max}\delta^2}{K(\delta,d)}. \tag{48}$$

8.2. Establishing Theorem 2

Using Theorem 1 and Lemma 8, we complete the proof of Theorem 2 under appropriate choice of algorithmic parameters. We start by setting some notation.

To that end, the algorithm as described in Section 4.1 iterates between MCTS and supervised learning. In particular, let $\ell \ge 1$ denote the iteration index. Let m_ℓ be the number of states that are sampled uniformly at random, independently, over $\mathcal{S}$ in this iteration, denoted as $S^{(\ell)} = \{s_i^{(\ell)} : i \in [m_\ell]\}$. Let $V^{(\ell-1)}$ be the input of value function from prior iteration; using this, the MCTS algorithm with n_ℓ simulations obtains improved estimates of value function for states in $S^{(\ell)}$ denoted as $\hat{V}^{(\ell)}\left(s_i^{(\ell)}\right)$, $i \in [m_\ell]$. Using $\left(s_i^{(\ell)}, \hat{V}^{(\ell)}\left(s_i^{(\ell)}\right)\right)_{i\in[m_\ell]}$, the nearest neighbor supervised learning as described previously with balls of appropriate radius $\delta_\ell \in (0,1)$ produces estimate $V^{(\ell)}$ for all states in $\mathcal{S}$. Let $\mathcal{F}^{(\ell)}$ denote the smallest σ-algebra containing all information pertaining to the algorithm (both MCTS and supervised learning). Define the error under MCTS in iteration ℓ as

$$\varepsilon_{\text{mcts}}^{(\ell)} = \mathbb{E}\left[\sup_{s\in\mathcal{S}}|\mathbb{E}\left[\hat{V}^{(\ell)}(s)|\mathcal{F}^{(\ell-1)}\right] - V^*(s)|\right]. \tag{49}$$

The error for supervised learning in iteration ℓ as

$$\theta_{\text{sl}}^{(\ell)} = \sup_{s\in\mathcal{S}}\left|V^{(\ell)}(s) - V^*(s)\right|, \quad \text{and} \quad \varepsilon_{\text{sl}}^{(\ell)} = \mathbb{E}\left[\theta_{\text{sl}}^{(\ell)}\right]. \tag{50}$$

Recall that in the beginning, we set $V^{(0)}(s) = 0$ for all $s \in \mathcal{S}$. Because $V^*(\cdot) \in [-V_{\max}, V_{\max}]$, we have that $\varepsilon_{\text{sl}}^{(0)} \le V_{\max}$. Furthermore, it is easy to see that, if the leaf estimates (i.e., the output of the supervised learning from the previous iteration) is bounded in $[-V_{\max}, V_{\max}]$, then the output of the MCTS algorithm is always bounded in $[-V_{\max}, V_{\max}]$. That is, because $V^{(0)}(s) = 0$ and the nearest neighbor supervised learning produces estimate $V^{(l)}$ via simple averaging, inductively, the output of the MCTS algorithm is always bounded in $[-V_{\max}, V_{\max}]$ throughout every iteration.

With the notation as previously set up, it follows that, for a given $\delta_\ell \in (0,1)$ with m_ℓ satisfying condition of Lemma 8, that is, $m_\ell = \Omega(d\delta_\ell^{-d-2}\log\delta_\ell^{-1})$, and with the nearest neighbor supervised learning using δ_ℓ radius

balls for estimation, we have the following recursion:

$$\varepsilon_{\mathrm{sl}}^{(\ell)} \le \varepsilon_{\mathrm{mcts}}^{(\ell)} + (C+1)\delta_\ell + \frac{4V_{\max}\delta_\ell^2}{K(\delta_\ell,d)} \le \varepsilon_{\mathrm{mcts}}^{(\ell)} + C'\delta_\ell, \quad (51)$$

where C' is a large enough constant, because $\frac{\delta_\ell^2}{K(\delta_\ell,d)} = \Theta\left(d\delta_\ell^{d+2}\right)$, which is $O(\delta_\ell)$ for all $\delta_\ell \in (0,1)$. By Theorem 1, for iteration $\ell+1$ that uses the output of supervised learning estimate, $V^{(\ell)}$, as the input to the MCTS algorithm, we obtain

$$\left|\mathbb{E}\left[\hat{V}^{(\ell+1)}(s)|\mathcal{F}^{(\ell)}\right] - V^*(s)\right| \le \gamma^{H^{(\ell+1)}}\mathbb{E}\left[\theta_{\mathrm{sl}}^{(\ell)}|\mathcal{F}^{(\ell)}\right] + O\left(n_{\ell+1}^{\eta-1}\right), \quad \forall s \in \mathcal{S}, \quad (52)$$

where $\eta \in [1/2,1)$ is the constant used by MCTS with fixed height of tree being $H^{(\ell+1)}$. This then implies that

$$\begin{aligned}\varepsilon_{\mathrm{mcts}}^{(\ell+1)} &= \mathbb{E}\left[\sup_{s\in\mathcal{S}}\left|\mathbb{E}\left[\hat{V}^{(\ell+1)}(s)|\mathcal{F}^{(\ell)}\right] - V^*(s)\right|\right] \\ &\le \gamma^{H^{(\ell+1)}}\mathbb{E}\left[\mathbb{E}\left[\theta_{\mathrm{sl}}^{(\ell)}|\mathcal{F}^{(\ell)}\right]\right] + O\left(n_{\ell+1}^{\eta-1}\right) \\ &\le \gamma^{H^{(\ell+1)}}\left(\varepsilon_{\mathrm{mcts}}^{(\ell)} + C'\delta_\ell\right) + O\left(n_{\ell+1}^{\eta-1}\right).\end{aligned} \quad (53)$$

Denote by $\lambda \triangleq \left(\frac{\varepsilon}{V_{\max}}\right)^{1/L}$. Because the final desired error ε should be less than $V_{\max}$ (otherwise, the problem is trivial by just outputing zero as the final estimates for all the states), we have $\lambda < 1$. Let us set the algorithmic parameters for MCTS and nearest neighbor supervised learning as follows: for each $\ell \ge 1$,

$$H^{(\ell)} = \left\lceil \log_\gamma \frac{\lambda}{8} \right\rceil, \delta_\ell = \frac{3V_{\max}}{4C'}\lambda^\ell, n_\ell = \kappa_l\left(\frac{8}{V_{\max}\lambda^\ell}\right)^{\frac{1}{1-\eta}}, \quad (54)$$

where $\kappa_l > 0$ is a sufficiently large constant such that $O\left(n_\ell^{\eta-1}\right) = \frac{V_{\max}}{8}\lambda^\ell$. Substituting these values into Equation (53) yields

$$\begin{aligned}\varepsilon_{\mathrm{mcts}}^{(\ell+1)} &= \mathbb{E}\left[\sup_{s\in\mathcal{S}}\left|\mathbb{E}\left[\hat{V}^{(\ell+1)}(s)|\mathcal{F}^{(\ell)}\right] - V^*(s)\right|\right] \\ &\le \frac{\lambda}{8}\varepsilon_{\mathrm{mcts}}^{(\ell)} + \frac{7V_{\max}}{32}\lambda^{\ell+1}.\end{aligned}$$

By (52) and (54), and the fact that $\varepsilon_{\mathrm{sl}}^{(0)} \le V_{\max}$, we have

$$\varepsilon_{\mathrm{mcts}}^{(1)} \le \frac{\lambda}{8}\varepsilon_{\mathrm{sl}}^{(0)} + \frac{\lambda}{8}V_{\max} \le \frac{\lambda}{4}V_{\max}.$$

It then follows inductively that

$$\varepsilon_{\mathrm{mcts}}^{(\ell)} \le \lambda^{\ell-1}\varepsilon_{\mathrm{mcts}}^{(1)} = \frac{V_{\max}}{4}\lambda^\ell.$$

As for the supervised learning oracle, $\forall s \in \mathcal{S}$, Equation (51) implies

$$\mathbb{E}\left[\sup_{s\in\mathcal{S}}\left|V^{(\ell)}(s) - V^*(s)\right|\right] \le \varepsilon_{\mathrm{mcts}}^{(\ell)} + \frac{3V_{\max}}{4}\lambda^\ell \le V_{\max}\lambda^\ell.$$

This implies that

$$\mathbb{E}\left[\sup_{s\in\mathcal{S}}\left|V^{(L)}(s) - V^*(s)\right|\right] \le V_{\max}\lambda^L = \varepsilon.$$

We now calculate the sample complexity, that is, the total number of state transitions required for the algorithm. During the ℓ th iteration, each query of MCTS oracle requires n_ℓ simulations. Recall that the number of querying MCTS oracle, that is, the size of training set $\mathcal{S}^{(\ell)}$ for the nearest neighbor supervised step, should satisfy $m_\ell = \Omega\left(d\delta_\ell^{-d-2}\log\delta_\ell^{-1}\right)$ (cf. Lemma 8). From Equation (54), we have

$$H^{(\ell)} = c_0'\log\lambda^{-1}, \quad \delta_\ell = c_1'\lambda^\ell, \text{ and } \quad n_\ell = c_2'\lambda^{-\ell/(1-\eta)},$$

where c_0', c_1', c_2', are constants independent of λ and ℓ. Each simulation of MCTS samples $H^{(\ell)}$ state transitions. Hence, the number of state transitions at the ℓ th iteration is given by

$$M^{(\ell)} = m_\ell n_\ell H^{(\ell)}.$$

Therefore, the total number of state transitions after L iterations is

$$\begin{aligned}\sum_{l=1}^{L} M^{(\ell)} &= \sum_{\ell=1}^{L} m_\ell \cdot n_\ell \cdot H^{(\ell)} \\ &= O\left(\varepsilon^{-\left(2+1/\left(1-\eta\right)+d\right)} \cdot \left(\log\frac{1}{\varepsilon}\right)^5\right).\end{aligned}$$

That is, for optimal choice of $\eta = 1/2$, the total number of state transitions is $O\left(\varepsilon^{-(4+d)} \cdot \left(\log\frac{1}{\varepsilon}\right)^5\right)$.

8.3. Proof of Lemma 8

Given N samples $s_i, i \in [N]$ that are sampled independently and uniformly at random over $\mathcal{S}$, and given the fact that each ball c_i, $i \in [K(\delta,d)]$ has at least $C_d\delta^d$ volume shared with $\mathcal{S}$, each of the sample falls within a given ball with probability at least $C_d\delta^d$. Let N_i, $i \in [K(\delta,d)]$ denote the number of samples among N samples in ball c_i.

Now the number of samples falling in any given ball is lower bounded by a Binomial random variable with parameter $N, C_d\delta^d$. By the Chernoff bound for the Binomial variable with parameter n, p, we have that

$$\mathbb{P}\left(B(n,p) \le np/2\right) \le \exp\left(-\frac{np}{8}\right).$$

Therefore, with an application of union bound, each ball has at least $0.5C_d\delta^d N$ samples with probability at least $1 - K(\delta,d)\exp\left(-C_d\delta^d N/8\right)$. That is, for $N = 32\max\left(1, \delta^{-2}V_{\max^2}\right)C_d^{-1}\delta^{-d}\left[\log\left(K(\delta,d) + \log\delta^{-1}\right]\right.$, each ball has

at least $\Gamma = 16\max\left(1, \delta^{-2}V_{\max}^2\right)\left(\log K(\delta,d) + \log\delta^{-1}\right)$ samples with probability at least $1 - \frac{\delta^2}{K(\delta,d)}$. Define event

$$\mathcal{E}_1 = \{N_i \geq 16\max\left(1, \delta^{-2}V_{\max}^2\right)\left(\log K(\delta,d) + \log\delta^{-1}\right), \quad \forall i \in [K(\delta,d)]\}.$$

Then

$$\mathbb{P}(\mathcal{E}_1^c) \leq \frac{\delta^2}{K(\delta,d)}.$$

Now, for any $s \in \mathcal{S}$, the nearest neighbor supervised learning described in Section 4.2 produces estimate $\hat{V}(s)$ equal to the average value of observations for samples falling in ball $c_{j(s)}$. Let $N_{j(s)}$ denote the number of samples in ball $c_{j(s)}$. To that end,

$$\begin{aligned}\left|\hat{V}(s) - V^*(s)\right| &= \left|\frac{1}{N_{j(s)}}\left(\sum_{i:s_i \in c_{j(s)}} V(s_i) - V^*(s)\right)\right| \\ &= \left|\frac{1}{N_{j(s)}}\left(\sum_{i:s_i \in c_{j(s)}} V(s_i) - \mathbb{E}[V(s_i)]\right)\right| + \left|\frac{1}{N_{j(s)}}\left(\sum_{i:s_i \in c_{j(s)}} \mathbb{E}[V(s_i)] - V^*(s_i)\right)\right| \\ &\quad + \left|\frac{1}{N_{j(s)}}\left(\sum_{i:s_i \in c_{j(s)}} V^*(s_i) - V^*(s)\right)\right|.\end{aligned}$$

For the first term, because for each $s_i \in c_{j(s)}$, $V(s_i)$ is produced using independent randomness via MCTS, and because the output $V(s_i)$ is a bounded random variable, using Hoeffding's inequality, it follows that

$$\mathbb{P}\left(\left|\frac{1}{N_{j(s)}}\left(\sum_{i:s_i \in c_{j(s)}} V(s_i) - \mathbb{E}[V(s_i)]\right)\right| \geq \Delta_1\right) \leq 2\exp\left(-\frac{N_{j(s)}\Delta_1^2}{8V_{\max^2}}\right).$$

The second term is no more than Δ because of the guarantee given by MCTS as assumed in the setup. Finally, the third term is no more than $C\delta$ because of Lipschitzness of V^*. To summarize, with probability at least $1 - 2\exp\left(-\frac{N_{j(s)}\Delta_1^2}{8V_{\max^2}}\right)$, we have that

$$|\hat{V}(s) - V^*(s)| \leq \Delta_1 + \Delta + C\delta.$$

As can be noticed, the algorithm produces the same estimate for all $s \in \mathcal{S}$ such that they map to the same ball. There are $K(\delta,d)$ such balls. Therefore, using union bound, it follows that with probability at least $1 - 2K(\delta,d)\exp\left(-\frac{\left(\min_{i\in[K(\delta,d)]}N_i\right)\Delta_1^2}{8V_{\max^2}}\right)$,

$$\sup_{s\in\mathcal{S}}\left|\hat{V}(s) - V^*(s)\right| \leq \Delta_1 + \Delta + C\delta.$$

Under event $\mathcal{E}_1$, $\min_{i\in[K(\delta,d)]}N_i \geq 16\max(1, \delta^{-2}V_{\max}^2)(\log K(\delta,d) + \log\delta^{-1})$. Therefore, under event $\mathcal{E}_1$, by choosing $\Delta_1 = \delta$, we have

$$\sup_{s\in\mathcal{S}}\left|\hat{V}(s) - V^*(s)\right| \leq \Delta + (C+1)\delta,$$

with probability at least $1 - \frac{2\delta^2}{K(\delta,d)}$. When event $\mathcal{E}_1$ does not hold or the previous expression does not hold, we have a trivial error bound of $2V_{\max}$ on the error. Therefore, we conclude that

$$\mathbb{E}\left[\sup_{s\in\mathcal{S}}\left|\hat{V}(s) - V^*(s)\right|\right] \leq \Delta + (C+1)\delta + \frac{4V_{\max}\delta^2}{K(\delta,d)}.$$

9. Conclusion

In this paper, we introduce a *correction* of the popular MCTS policy for improved value function estimation for a given state, using an existing value function estimation for the entire state space. This correction was obtained through careful, rigorous analysis of a nonstationary MAB where rewards are dependent and nonstationary. In particular, we analyzed a variant of the classical UCB policy for such an MAB. Using this as a building block, we establish rigorous performance guarantees for the *corrected* version of MCTS proposed in this work. This, to the best of our knowledge, is the first mathematically correct analysis of the UCT policy despite its popularity since it has been proposed in literature (Kocsis and Szepesvári 2006, Kocsis et al. 2006). We further establish that the proposed MCTS policy, when combined with nearest neighbor supervised learning, leads to near optimal sample complexity for obtaining estimation of value function within a given tolerance, where the optimality is in the minimax sense. This suggests the tightness of our analysis and the utility of the MCTS policy.

Much of this work was inspired by the success of AGZ that uses MCTS combined with supervised learning. Interestingly enough, the correction of MCTS suggested by our analysis is qualitatively similar to the version of MCTS used by AGZ as reported in practice. This seeming coincidence may suggest further avenue for practical utility of versions of the MCTS proposed in this work and is an interesting direction for future work.

Appendix A. Extension of Theorem 1 for Stochastic Environment

We established Theorem 1 when the transition kernel is deterministic. We now explain how to extend the results to the setting with stochastic transition kernel. We do so by effectively mapping the stochastic setting to a deterministic setting as discussed next.

We start by defining the stochastic environment. Recall that when an action a is taken at state s, the next state is s' with probability $\mathcal{P}(s'|s,a)$. In the deterministic setting, we have $\mathcal{P}(s'|s,a) \in \{0,1\}$, whereas in the stochastic setting, we allow for $\mathcal{P}(s'|s,a) \in [0,1]$. We further consider the following setup. Let there be a fixed $\phi > 0$ so that

$$\inf\{\mathcal{P}(s'|s,a): \mathcal{P}(s'|s,a) \neq 0,\ s,s' \in \mathcal{S}, a \in \mathcal{A}\} \geq \phi. \tag{A.1}$$

Let $\text{supp}(s,a)$ be the support of the distribution $\mathcal{P}(\cdot|s,a)$. Because of (A.1), $|\text{supp}(s,a)| \leq \lfloor\frac{1}{\phi}\rfloor \equiv M$. That is, the number

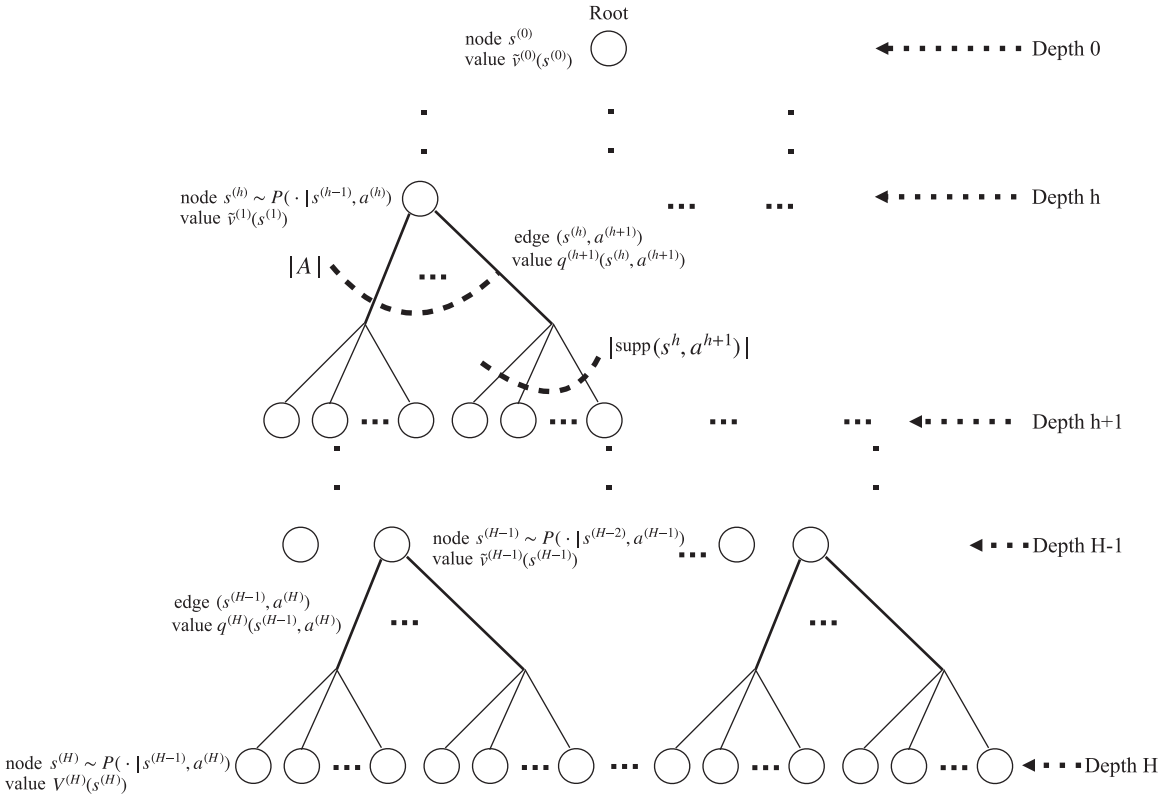
of next state reachable for a given state s under an action a is bounded by a constant M for all s, a.

Let us consider the MCTS algorithm for such a stochastic setting. At a node (i.e., state) at depth h, the action with the highest sum of average reward and a polynomial bonus is selected. A next state at depth $h+1$ is reached, and the process is repeated until a fix depth H. We then update the corresponding statistics of the nodes and the selected actions at each depth, and this finishes one iteration of the simulation. Because the transitions are stochastic, for state (node) s at each depth, each action $a \in \mathcal{A}$ would have up to $|\text{supp}(s,a)| \le M$ children nodes. In contrast, for the deterministic case, each action leads to a unique state at the next depth (as shown in Figure 1, where each edge represents an action and connects a node s at depth h to a unique next state s' at depth $h+1$). However, despite of the distinct difference, we can map the stochastic scenario back to the deterministic setting via a simple transformation. Specifically, given the state s at depth h and action a, although there are multiple next states, for the purpose of MCTS decision, we assign a "meta-edge" corresponding to each action $a \in \mathcal{A}$ for a given state $s \in \mathcal{S}$. This edge connects s via action a to all of its next states in $\text{supp}(s,a)$. This is illustrated in Figure A.1, where each thick edge is a meta-edge representing an action in $\mathcal{A}$.

In the deterministic setting, at the end of each simulation step, the rewards of nodes and edges were updated along the entire path visited in the simulation step as described in Algorithm 1. In the stochastic setting, we perform the same operation, that is, updating the rewards for each node (state) and each action (i.e., the meta-edge) in the same manner. Now we might have a larger tree because of multiple children associated with the same action for a given state. Finally, while similar in spirit, the key difference lies in how we selection an action $a \in \mathcal{A}$ at a given state $s \in \mathcal{S}$ at depth h of the tree in a simulation step. In the deterministic setting, we simply use the sum of the empirical average return and the polynomial bonus term associated with the action (or the edge), as described in (5). In the stochastic setting, for each action a at a state s, instead, we use a weighted sum of the empirical average returns associated with all possible next states, with weights simply being the empirical frequency of visiting each next state in $\text{supp}(s,a)$ thus far. We use a similar polynomial bonus term for each action.

With the modifications elaborated previously, we can then reuse the majority of our previous analysis. Recall that to establish the desired theorem for MCTS with deterministic transitions, we recursively argue the convergence and polynomial concentration properties at each depth. That is, starting with the convergence and concentration properties for nodes at depth $h+1$, we show the convergence and concentration properties for nodes at depth h and then recursively apply this process until we reach the root node. More precisely, the induction step is completed by analyzing a nonstationary MAB problem where the (nonstationary) outcomes of each arm converge and polynomially concentrate. In the stochastic setting, the algorithm dynamics are almost the same as that for deterministic setting, except that on taking an arm (action),

Figure A.1. MCTS with Stochastic Transitions



there is additional randomness determining which children in $\text{supp}(s,a)$ we transition to. Suppose that we can argue that the nonstationary outcomes of each arm, after accounting for the stochastic transition through weighted average with empirical frequency, have the same convergence and polynomially concentration properties as the children nodes. Consequently, we can apply the analysis we developed for the deterministic case by following the same line of induction argument.

Specifically, we can reduce the analysis of MCTS for stochastic settings to that of the deterministic settings as shown in Figure A.2. We view the children nodes associated with one action collectively as one meta-node corresponding to the action, that is, the meta-node encapsulates the randomness of the transitions and the nonstationary reward processes at the children nodes. At depth $h+1$, starting with the convergence and concentration properties for the nonstationary reward processes at each child node, we show that the reward process at the meta-node has the same convergence and concentration properties. The action selection problem at each node/state for the stochastic setting then is reduced to the MAB problem we analyzed in the deterministic setting, for which we have established the convergence and concentration properties for the parent nodes at depth h. By following the proof for the deterministic settings, we shall obtain the guarantees for MCTS with stochastic environments. To summarize, it is clear that to establish the desired results for MCTS, we only need to fill in the missing step of arguing the convergence and concentration properties of the meta-node; the rest of the proof then exactly follows without modifications.

To this end, we consider a mathematical formulation that precisely describes the action selection problem at a node with stochastic transition. Consider a multinomial distribution over $[M]=\{1,\dots,M\}$ with $p_m \ge \phi$ being probability of observing outcome $m \in [M]$. We denote the distribution by $\text{Dist}(p)$. Let us consider a sequence of i.i.d. random variables $\{Y_i, i \in \mathbb{N}^+\}$, where $Y_i \sim \text{Dist}(p)$. Consider M random processes (possibly dependent) $\{X_{m,t}, t \in \mathbb{N}^+\}$ for $1 \le m \le M$. Define a random process $\{Z_i, i \in \mathbb{N}^+\}$ as follows: $Z_i = \sum_{m=1}^{M} \mathbb{I}\{Y_i = m\} X_{m,N(m,i-1)+1}$, where $N(m,i-1) = \sum_{j=1}^{i-1} \mathbb{I}\{Y_j = m\}$ is the total number of times that the mth outcome has been generated up to (and including) time $i-1$. In the context of MAB with stochastic transition, the introduced random processes are associated with one arm a as follows: playing action a leads to a random next state in $[M]$ according to $\text{Dist}(p)$; state $m \in [M]$ is associated with a reward sequence $\{X_{m,t}, t \in \mathbb{N}^+\}$; Z_i represents the reward obtained by playing the action a for the ith time. We establish that if for each $m \in [M]$, the random process $\{X_{m,t}, t \in \mathbb{N}^+\}$ satisfies a convergence and the polynomial concentration properties, then so does the random process $\{Z_i\}$, as stated in the following lemma.

Lemma A.1. *Suppose that the M random processes $\{X_{m,t}, t \in \mathbb{N}^+\}$, $1 \le m \le M$, satisfy*

A. *Convergence: for $n \ge 1$, with notation $\bar{X}_{m,n} = \frac{1}{n}\left(\sum_{t=1}^{n} X_{m,t}\right)$,*

$$\lim_{n\to\infty} \mathbb{E}\left[\bar{X}_{m,n}\right] = \mu_m, \quad \forall 1 \le m \le M.$$

B. *Concentration: there exist constants, $\beta > 1$, $\xi > 1$, $1/2 \le \eta < 1$ such that for $n \ge 1$ and $z \ge 1$,*

$$\mathbb{P}\left(n\bar{X}_{m,n} - n\mu_m \ge n^{\eta} z\right) \le \frac{\beta}{z^{\xi}},$$

$$\mathbb{P}\left(n\bar{X}_{m,n} - n\mu_m \le -n^{\eta} z\right) \le \frac{\beta}{z^{\xi}}, \quad \forall 1 \le m \le M.$$

Then, the random process $\{Z_i, i \in \mathbb{N}^+\}$ satisfies

A. *Convergence: for $n \ge 1$, with notation $\bar{Z}_n = \frac{1}{n}\left(\sum_{i=1}^{n} Z_i\right)$,*

$$\lim_{n\to\infty} \mathbb{E}\left[\bar{Z}_n\right] = \sum_{m=1}^{M} p_m \mu_m.$$

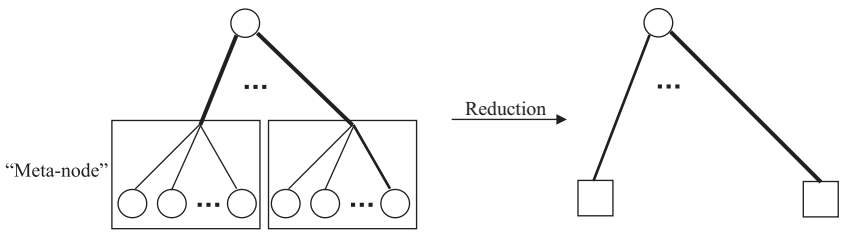
B. *Concentration: there exist constant $\beta' > 1$ depending upon M, ξ, β such that for $n \ge 1$ and $z \ge 1$,*

$$\mathbb{P}\left(n\bar{Z}_n - n\left(\sum_{m=1}^{M} p_m \mu_m\right) \ge n^{\eta} z\right) \le \frac{\beta'}{z^{\xi}},$$

$$\mathbb{P}\left(n\bar{Z}_n - n\left(\sum_{m=1}^{M} p_m \mu_m\right) \le -n^{\eta} z\right) \le \frac{\beta'}{z^{\xi}}.$$

As discussed, with Lemma A.1, the proof in the main paper is then readily extended to the stochastic setting. One important aspect that is worth mentioning is that the constants related to the polynomial rate, η and ξ, are preserved and remain unchanged from the processes $\{X_{m,\cdot}\}$ to the process Z, that is, the meta-nodes has the same polynomial rate as the children nodes. Only the constant β is different. This means that the proof of Theorem 1 can be applied with a simple change of a different constant β'. Particularly, Theorem 1 holds with the same rate of convergence, that is, $O(n^{\eta-1})$. Finally, one may notice that in Lemma A.1, for the concentration of $\{X_{m,t}, t \in \mathbb{N}^+\}$, we assume $\xi > 1$ instead of a more general choice $\xi > 0$ (cf., Section 5). This is indeed not an issue, as one can easily verify that the conditions in Theorem 1, that is, choosing a large $\xi^{(H)}$ at depth H and using the algorithmic choices (6)–(8), implicitly guarantees $\xi > 1$ for every depth recursively.

Figure A.2. Reduce the Stochastic Transitions to a Single "Meta-Node" for Each Action



A.1. Proof of Lemma A.1

Fix n. Note that according to the generating process, we can rewrite $\bar{Z}_n$ as

$$\bar{Z}_n = \frac{1}{n}\left(\sum_{m=1}^{M}\sum_{i=1}^{\mathcal{N}_m} X_{m,i}\right),$$

where $\mathcal{N}_m, 1 \le m \le M$ are random variables such that $\sum_{m=1}^{M}\mathcal{N}_m = n$ and $\mathcal{N}_m \sim \text{Binomial}(n, p_m)$, that is, $\mathcal{N}_m$ is the number of times the mth outcome is generated according to the distribution Dist(p) after n trials. By Hoeffding's inequality, we have that for $1 \le m \le M$ and $t \ge 0$,

$$\mathbb{P}(\mathcal{N}_m\mu_m - np_m\mu_m \ge t) \le \exp\left(-\frac{2t^2}{n\mu_m^2}\right).$$

Therefore,

$$\mathbb{P}(\mathcal{N}_m\mu_m - np_m\mu_m \ge p_m n^{\eta} z) \le \exp\left(-\frac{2p_m^2 z^2 n^{2\eta-1}}{\mu_m^2}\right) \le \frac{\beta_m}{z^{\xi}},$$

where β_m is a large enough constant depending on ξ, p_m, and μ_m and importantly, independent of n. The last step follows because the exponential tail resulted from the Hoeffding's inequality decays faster than a polynomial one. We have that

$$\mathbb{P}\left(n\bar{Z}_n - \sum_{m=1}^{M} np_m\mu_m \ge n^{\eta} z\right) \le \mathbb{P}\left(n\bar{Z}_n - \sum_{m=1}^{M} np_m\mu_m \ge \sum_{m=1}^{M}\frac{\mathcal{N}_m^{\eta} z}{2M} + \sum_{m=1}^{M}\frac{p_m n^{\eta} z}{2}\right), \tag{A.2}$$

$$\begin{aligned} &= \mathbb{P}\left(\sum_{m=1}^{M}\mathcal{N}_m\bar{X}_{m,\mathcal{N}_m} - \sum_{m=1}^{M} np_m\mu_m \ge \sum_{m=1}^{M}\frac{\mathcal{N}_m^{\eta} z}{2M} + \sum_{m=1}^{M}\frac{p_m n^{\eta} z}{2}\right), \\ &\le \sum_{m=1}^{M}\mathbb{P}\left(\mathcal{N}_m\bar{X}_{m,\mathcal{N}_m} - np_m\mu_m \ge \frac{\mathcal{N}_m^{\eta} z}{2M} + \frac{p_m n^{\eta} z}{2}\right). \end{aligned} \tag{A.3}$$

Note that (A.2) follows because the following holds almost surely:

$$\sum_{m=1}^{M}\frac{\mathcal{N}_m^{\eta} z}{2M} + \sum_{m=1}^{M}\frac{p_m n^{\eta} z}{2} \le \sum_{m=1}^{M}\frac{n^{\eta} z}{2M} + \sum_{m=1}^{M}\frac{p_m n^{\eta} z}{2} = n^{\eta} z.$$

Furthermore, (A.2) holds because

$$\begin{aligned} \mathbb{P}(A + B \ge C + D) &\le \mathbb{P}(A \ge C \text{ or } B \ge D) \\ &\le \mathbb{P}(A \ge C) + \mathbb{P}(B \ge D). \end{aligned}$$

To continue, we have that

$$\begin{aligned} &\mathbb{P}\left(\mathcal{N}_m\bar{X}_{m,\mathcal{N}_m} - np_m\mu_m \ge \frac{\mathcal{N}_m^{\eta} z}{2M} + \frac{p_m n^{\eta} z}{2}\right) \\ &= \mathbb{P}\left(\mathcal{N}_m\bar{X}_{m,\mathcal{N}_m} - \mathcal{N}_m\mu_m + \mathcal{N}_m\mu_m - np_m\mu_m \ge \frac{\mathcal{N}_m^{\eta} z}{2M} + \frac{p_m n^{\eta} z}{2}\right) \\ &\le \mathbb{P}\left(\mathcal{N}_m\bar{X}_{m,\mathcal{N}_m} - \mathcal{N}_m\mu_m \ge \frac{\mathcal{N}_m^{\eta} z}{2M}\right) + \mathbb{P}\left(\mathcal{N}_m\mu_m - np_m\mu_m \ge \frac{p_m n^{\eta} z}{2}\right) \\ &= \mathbb{E}\left[\mathbb{P}\left(\mathcal{N}_m\bar{X}_{m,\mathcal{N}_m} - \mathcal{N}_m\mu_m \ge \frac{\mathcal{N}_m^{\eta} z}{2M}\middle|\mathcal{N}_m\right)\right] \\ &\quad + \mathbb{P}\left(\mathcal{N}_m\mu_m - np_m\mu_m \ge \frac{p_m n^{\eta} z}{2}\right) \\ &\le \mathbb{E}\left[\frac{\beta(2M)^{\xi}}{z^{\xi}}\right] + \frac{2^{\xi}\beta_m}{z^{\xi}} \\ &\le \frac{\beta'_m}{z^{\xi}}, \end{aligned} \tag{A.4}$$

where $\beta'_m = \beta(2M)^{\xi} + 2^{\xi}\beta_m$. Note that $\mathbb{P}\left(\mathcal{N}_m\bar{X}_{m,\mathcal{N}_m} - \mathcal{N}_m\mu_m \ge \frac{\mathcal{N}_m^{\eta} z}{2M}\middle|\mathcal{N}_m\right) \le \frac{\beta(2M)^{\xi}}{z^{\xi}}$ holds, because if $z \ge 2M$, the concentration inequality for $\{\bar{X}_{m,\cdot}\}$ assumed in the lemma applies; and if $1 \le z < 2M$, the R.H.S. of the previous inequality is larger than one because $\beta > 1$ and the inequality trivially holds. Combining (A.3) and (A.4), we have that

$$\mathbb{P}\left(n\bar{Z}_n - \sum_{m=1}^{M} np_m\mu_m \ge n^{\eta} z\right) \le \sum_{m=1}^{M}\frac{\beta'_m}{z^{\xi}} \le \frac{\beta'}{z^{\xi}},$$

where $\beta' = M\max_{1\le m\le M}\beta'_m$. The other side of the inequality follows similarly, and this completes the proof of the desired concentration property of $\bar{Z}_n$.

For convergence, note that we have established the concentration property that for $z \ge 1$:

$$\mathbb{P}\left(\left|\bar{Z}_n - \sum_{m=1}^{M} p_m\mu_m\right| \ge n^{\eta-1} z\right) \le \frac{2\beta'}{z^{\xi}}.$$

Therefore,

$$\begin{aligned} \mathbb{E}\left[\left|\bar{Z}_n - \sum_{m=1}^{M} p_m\mu_m\right|\right] &= \int_0^{\infty}\mathbb{P}\left(\left|\bar{Z}_n - \sum_{m=1}^{M} p_m\mu_m\right| \ge s\right)ds \\ &= \int_0^{n^{\eta-1}}\mathbb{P}\left(\left|\bar{Z}_n - \sum_{m=1}^{M} p_m\mu_m\right| \ge s\right)ds \\ &\quad + \int_{n^{\eta-1}}^{\infty}\mathbb{P}\left(\left|\bar{Z}_n - \sum_{m=1}^{M} p_m\mu_m\right| \ge s\right)ds \\ &\le n^{\eta-1} + \int_{n^{\eta-1}}^{\infty}\frac{2\beta' n^{\xi(\eta-1)}}{s^{\xi}}ds \\ &= n^{\eta-1} + \frac{2\beta' n^{\eta-1}}{\xi - 1}, \end{aligned}$$

where the integral is finite because $\xi > 1$ by assumption in the lemma. Therefore,

$$\begin{aligned} \lim_{n\to\infty}\left|\mathbb{E}\left[\bar{Z}_n - \sum_{m=1}^{M} p_m\mu_m\right]\right| &\le \lim_{n\to\infty}\mathbb{E}\left[\left|\bar{Z}_n - \sum_{m=1}^{M} p_m\mu_m\right|\right] \\ &\le \lim_{n\to\infty}\left(n^{\eta-1} + \frac{2\beta' n^{\eta-1}}{\xi - 1}\right) = 0. \end{aligned}$$

The limit is zero because $1/2 \le \eta < 1$. The previous expression implies that $\lim_{n\to\infty}\mathbb{E}[\bar{Z}_n] = \sum_{m=1}^{M} p_m\mu_m$, which establishes the desired convergent property of $\bar{Z}_n$. This completes the proof of Lemma A.1.

Appendix B. Numerical Experiments

Although the focus of this paper is to develop a theoretical understanding of MCTS, we provide simple toy examples as supplements to corroborate our results. To this end, we design a simple class of deterministic MDPs as follows. For each state $s \in \mathcal{S}$ and each action $a \in \mathcal{A}$, we sample uniformly from $\mathcal{S}$ a state and fix it to be the corresponding next state s'. The reward $\mathcal{R}(s,a)$ is a uniformly distributed random variable taking values in $[0, R_{\max}(s,a)]$, where the bound $R_{\max}(s,a)$ is uniformly sampled from the interval $[-3,3]$ beforehand and is then fixed. We let $|\mathcal{S}| = 20$, $|\mathcal{A}| = 5$ and $\gamma = 0.8$. We then sample a deterministic MDP from the previous class and query a state via the MCTS algorithm with different depth H. For selecting an action at each depth, we use the polynomial bonus term

(Eq. (5)) as emphasized throughout the paper with $\eta = 1/2$. That is, we choose the action with the highest upper confidence bound in the form of "mean reward + $C \cdot t_s^{1/4}/t_a^{1/2}$." Here, t_s is the number of times that particular node at depth h has been visited; t_a is the number of times the action a is chosen for that node; and C is a constant for controlling exploration and exploitation. For simplicity, we choose the same C for each depth as this is common in practice. The value of the leaf nodes is set to zero. Per our theoretical results (Theorem 1 and Section 7.5), the output of the MCTS algorithm, in expectation, converges to the value estimate after running H steps of value function iteration starting with $\hat{V} \equiv 0$ for all states. To validate this consistency result, we perform 25 independent queries of MCTS with a selected root state and plot the resulting mean and standard deviation. The value estimate after H steps of value function iteration is used as the "true value" to benchmark the experiments. Figure B.1 shows the results for two tree depths: $H = 7$ (left) and $H = 10$ (right). As expected, the output of MCTS converges to the desired true value. The constant C captures the extent of the exploration-exploitation tradeoff. With smaller C, the simulation could be underexplored and the error bars are wider because of occasionally inaccurate estimates for some runs. A larger C implies more exploration; consequently, it requires more simulation steps to converge. We note that $C = 1$ seems to be a good choice in this example.

Because our results can be extended to the stochastic environments, we also experiment with stochastic MDPs. The class of stochastic MDPs is constructed in the same manner as before except that for each state $s \in \mathcal{S}$ and each action $a \in \mathcal{A}$: (1) we sample L states uniformly from $\mathcal{S}$ and fix them to be the potential next states; and (2) the transition kernel $\mathcal{P}(\cdot|s,a)$ is then sampled from a Dirichlet distribution with L categories. We let $|\mathcal{S}| = 100$, $|\mathcal{A}| = 3$, $L = 3$, and $\gamma = 0.8$. A stochastic MDP is then sampled from the class, and we again perform 25 independent MCTS queries with different depth H. Figure B.2 summarizes the corresponding results. A large number of simulation steps is used to account for the additional stochasticity from the transition. Again, these experiments corroborate our theoretical findings, with the mean of the outputs converging to the true value.

Appendix C. Proof of Proposition 1

Recent work (Shah and Xie 2018) establishes a lower bound on the sample complexity for reinforcement learning algorithms on MDPs. We follow a similar argument to establish a lower bound on the sample complexity for MDPs with deterministic transitions. We provide the proof for completeness. The key idea is to connect the problem of estimating the value function to the problem of nonparametric regression and then leveraging known minimax lower bound for the latter. In particular, we show that a class of nonparametric regression problems can be embedded in an MDP with deterministic transitions, so any algorithm for the latter can be used to solve the former. Prior work on nonparametric regression (Stone 1982, Tsybakov 2009) establishes that a certain number of observations is *necessary* to achieve a given accuracy using *any* algorithms, hence leading to a corresponding necessary condition for the sample size of estimating the value function in an MDP problem. We now provide the details.

Step 1. Nonparametric Regression. Consider the following nonparametric regression problem: Let $\mathcal{S} := [0,1]^d$ and assume that we have T data pairs $(x_1, y_1), \ldots, (x_T, y_T)$ such that conditioned on $x_1, \ldots, x_n$, the random variables $y_1, \ldots, y_n$ are independent and satisfy

$$\mathbb{E}[y_t | x_t] = f(x_t), \qquad x_t \in \mathcal{S}, \tag{C.1}$$

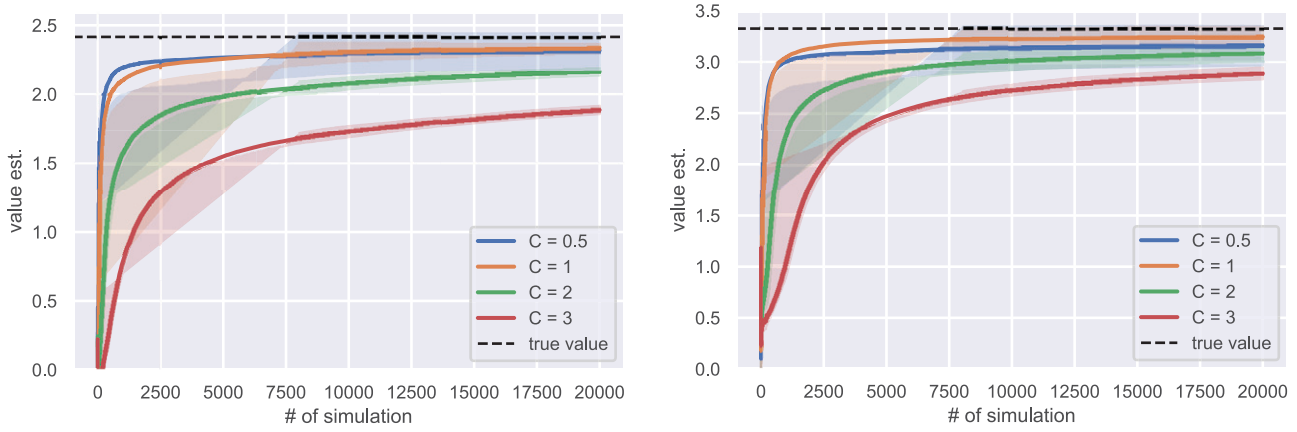
where $f : \mathcal{S} \to \mathbb{R}$ is the unknown regression function. Suppose that the conditional distribution of y_t given $x_t = x$ is a Bernoulli distribution with mean $f(x)$. We also assume that f is $1-$ Lipschitz continuous with respect to the Euclidean norm, that is,

$$|f(x) - f(x_0)| \le |x - x_0|, \quad \forall\, x, x_0 \in \mathcal{S}.$$

Let $\mathcal{F}$ be the collection of all $1-$ Lipschitz continuous function on $\mathcal{X}$, that is,

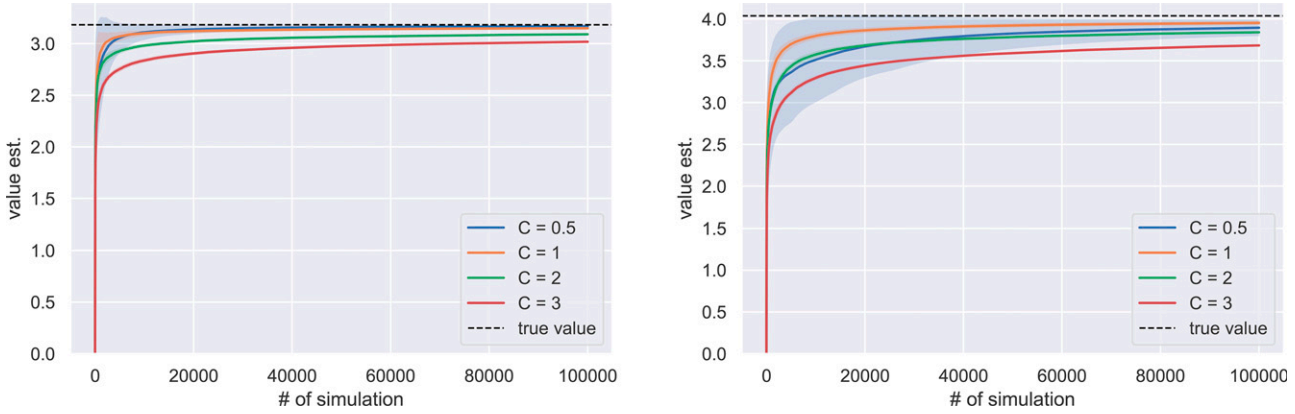
$$\mathcal{F} = \{h | h \text{ is a 1-Lipschitz function on } \mathcal{S}\},$$

Figure B.1. (Color online) Simulation for a Deterministic MDP with Tree Depth $H = 7$ (Left) and $H = 10$ (Right)



Note. Each line is a summary of 25 MCTS experiments showing the mean and standard deviation.

Figure B.2. (Color online) Simulation for a Stochastic MDP with Tree Depth $H = 5$ (Left) and $H = 8$ (Right)



Note. Each line is a summary of 25 MCTS experiments showing the mean and standard deviation.

The goal is to estimate f given the observations $(x_1, y_1), \ldots, (x_T, y_T)$ and the prior knowledge that $f \in \mathcal{F}$.

It is easy to verify that the previous problem is a special case of the nonparametric regression problem considered in the work by Stone (1982) (in particular, example 2 therein). Let $\hat{f}_T$ denote an arbitrary (measurable) estimator of f based on the training samples $(x_1, y_1), \ldots, (x_T, y_T)$. By theorem 1 in Stone (1982), we have the following result: there exists a $c > 0$, such that

$$\lim_{T\to\infty} \inf_{\hat{f}_T} \sup_{f\in\mathcal{F}} \mathbb{P}\left(\|\hat{f}_T - f\|_\infty \geq c\left(\frac{\log T}{T}\right)^{\frac{1}{2+d}}\right) = 1,$$

where infimum is over all possible estimators $\hat{f}_T$. Translating this result to the nonasymptotic regime, we obtain the following theorem.

Theorem C.1. *Under the previously stated assumptions, for each $\delta \in (0,1)$, there exists $c > 0$ and T_δ such that*

$$\inf_{\hat{f}_T} \sup_{f\in\mathcal{F}} \mathbb{P}\left(\|\hat{f}_T - f\|_\infty \geq c\left(\frac{\log T}{T}\right)^{\frac{1}{2+d}}\right) \geq \delta, \qquad \text{for all } T \geq T_\delta.$$

Step 2. MDP with Deterministic Transitions. Consider a class of discrete-time discounted MDPs $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where

$\mathcal{S} = [0,1]^d$,
$\mathcal{A}$ is finite,
for each (x,a), there exists a unique $x' \in \mathcal{S}$ such that $\mathcal{P}(x' \mid x, a) = 1$,
$r(x,a) = r(x)$ for all a,
$\gamma = 0$.

In words, the transition is deterministic, the expected reward is independent of the action taken and the current state, and only immediate reward matters.

Let R_t be the observed reward at step t. We assume that given x_t, the random variable R_t is independent of $(x_1, \ldots, x_{t-1})$, and follows a Bernoulli distribution Bernoulli $(r(x_t))$. The expected reward function $r(\cdot)$ is assumed to be 1 − Lipschitz and bounded. It is easy to see that for all $x \in \mathcal{S}$, $a \in \mathcal{A}$,

$$V^*(x) = r(x). \tag{C.2}$$

Step 3. Reduction from Regression to MDP. Given a nonparametric regression problem as described in Step 1, we may reduce it to the problem of estimating the value function V^* of the MDP described in Step 2. To do this, we set

$$r(x) = f(x), \qquad \forall x \in \mathcal{S}$$

and

$$R_t = y_t, \qquad t = 1, 2, \ldots, T.$$

In this case, it follows from Equations (C.2) that the value function is given by $V^* = f$. Moreover, the expected reward function $r(\cdot)$ is 1 − Lipschitz, so the assumptions of the MDP in Step 2 are satisfied. This reduction shows that the MDP problem is at least as hard as the nonparametric regression problem, so a lower bound for the latter is also a lower bound for the former.

Applying Theorem C.1 yields the following result: for any number $\delta \in (0,1)$, there exist some numbers $c > 0$ and $T_\delta > 0$, such that

$$\inf_{\hat{V}_T} \sup_{V^*\in\mathcal{F}} \mathbb{P}\left[\|\hat{V}_T - V^*\|_\infty \geq c\left(\frac{\log T}{T}\right)^{\frac{1}{2+d}}\right] \geq \delta, \qquad \text{for all } T \geq T_\delta.$$

Consequently, for any reinforcement learning algorithm $\hat{V}_T$ and any sufficiently small $\varepsilon > 0$, there exists an MDP problem with deterministic transitions such that, to achieve

$$\mathbb{P}\left[\|\hat{V}_T - V^*\|_\infty < \varepsilon\right] \geq 1 - \delta,$$

one must have

$$T \geq C' d \left(\frac{1}{\varepsilon}\right)^{2+d} \log\left(\frac{1}{\varepsilon}\right),$$

where $C' > 0$ is a constant. The statement of Proposition 1 follows by selecting $\delta = \frac{1}{2}$.

Endnote

[1] We use the standard notation $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ to hide logarithmic terms in the big-O and big-Ω asymptotic notation.

References

Agrawal R (1995) Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Adv. Appl. Probability* 27(4):1054–1078.

Audibert JY, Munos R, Szepesvári C (2009) Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Comput. Sci.* 410(19):1876–1902.

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learn.* 47(2-3): 235–256.

Auger D, Couetoux A, Teytaud O (2013) Continuous upper confidence trees with polynomial exploration–consistency. *Proc. Joint Eur. Conf. on Machine Learn. and Knowledge Discovery in Databases* (Springer, Berlin), 194–209.

Azizzadenesheli K, Yang B, Liu W, Brunskill E, Lipton ZC, Anandkumar A (2018) Sample-efficient deep RL with generative adversarial tree search. Preprint, version 4, submitted September 5, 2019, arXiv:1806.05780.

Bartlett P, Gabillon V, Healey J, Valko M (2019) Scale-free adaptive planning for deterministic dynamics & discounted rewards. Chaudhuri K, Salakhutdinov R, eds. *Proc. 36th Internat. Conf. on Machine Learn.*, vol. 97. Proceedings of Machine Learning Research (PMLR), 495–504.

Bertsekas D (1975) Convergence of discretization procedures in dynamic programming. *IEEE Trans. Automated Control* 20(3):415–419.

Bertsekas D (2017) *Dynamic Programming and Optimal Control* (Athena Scientific).

Browne CB, Powley E, Whitehouse D, Lucas SM, Cowling PI, Rohlfshagen P, Tavener S, et al. (2012) A survey of Monte Carlo tree search methods. *IEEE Trans. Comput. Intelligent AI Games* 4(1):1–43.

Chang HS, Fu MC, Hu J, Marcus SI (2005) An adaptive sampling algorithm for solving markov decision processes. *Oper. Res.* 53(1): 126–139.

Coquelin PA, Munos R (2007) Bandit algorithms for tree search. Preprint, version 1, submitted March 13, https://arxiv.org/abs/0703062.

Coulom R (2006) Efficient selectivity and backup operators in monte-carlo tree search. *Proc. Internat. Conf. on Comput. and Games* (Springer, Berlin), 72–83.

Dufour F, Prieto-Rumeau T (2012) Approximation of Markov decision processes with general state space. *J. Math. Anal. Appl.* 388 (2):1254–1267.

Dufour F, Prieto-Rumeau T (2013) Finite linear programming approximations of constrained discounted Markov decision processes. *SIAM J. Control Optim.* 51(2):1298–1324.

Efroni Y, Dalal G, Scherrer B, Mannor S (2018) Multiple-step greedy policies in approximate and online reinforcement learning. Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 31 (Curran Associates, Inc.), 5244–5253.

Even-Dar E, Mansour Y, Bartlett P (2003) Learning rates for Q-learning. *J. Machine Learn. Res.* 5(1).

Guo X, Singh S, Lee H, Lewis RL, Wang X (2014) Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning. Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, vol. 27 (Curran Associates, Inc.), 3338–3346.

Hoeffding W (1963) Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* 25:13–30.

Hren JF, Munos R (2008) Optimistic planning of deterministic systems. *Proc. Eur. Workshop on Reinforcement Learn.* (Springer, Berlin), 151–164.

Jiang DR, Ekwedike E, Liu H (2018) Feedback-based tree search for reinforcement learning. *Proc. Internat. Conf. on Machine Learn.*

Kakade S (2003) On the sample complexity of reinforcement learning. PhD thesis, University of London, University College London.

Kaufmann E, Koolen WM (2017) Monte-carlo tree search by best arm identification. Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Inc.), 4897–4906.

Kearns M, Mansour Y, Ng AY (2002) A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learn.* 49(2-3):193–208.

Kocsis L, Szepesvári C (2006) Bandit based Monte-Carlo planning. *Proc. Eur. Conf. on Machine Learn.* (Springer, Berlin), 282–293.

Kocsis L, Szepesvári C, Willemson J (2006) Improved Monte-Carlo search. Technical report, University of Tartu, Tartu, Estonia.

Mao W, Zhang K, Xie Q, Basar T (2020) Poly-hoot: Monte-carlo planning in continuous space mdps with non-asymptotic analysis. *Adv. Neural Inform. Processing Systems* 33:4549–4559.

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. Preprint, version 1, submitted December 19, https://arxiv.org/abs/1312.5602.

Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, et al. (2016) Asynchronous methods for deep reinforcement learning. *Proc. Internat. Conf. on Machine Learn* 1928–1937.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, et al (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529.

Munos R (2014) From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations Trends® Machine Learn.* 7(1):1–129.

Salomon A, Audibert JY (2011) Deviations of stochastic bandit regret. *Proc. Internat. Conf. on Algorithmic Learn. Theory* (Springer, Berlin), 159–173.

Schadd MPD, Winands MHM, van den Herik HJ, Chaslot GMJB, Uiterwijk JWHM (2008) Single-player Monte-Carlo tree search. van den Herik HJ, Xu X, Ma Z, Winands MHM, eds. *Computers and Games* (Springer, Berlin), 1–12.

Schulman J, Levine S, Abbeel P, Jordan M, Moritz P (2015) Trust region policy optimization. *Proc. Internat. Conf. on Machine Learn.* 1889–1897.

Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. Preprint, version 2, submitted August 28, https://arxiv.org/abs/1707.06347.

Shah D, Xie Q (2018) Q-learning with nearest neighbors. Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 31 (Curran Associates), 3115–3125.

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, et al. (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, et al (2017a) Mastering chess and shogi by self-play with a general reinforcement learning algorithm. Preprint, version 1, submitted December 5, 2017, https://arxiv.org/abs/1712.01815.

Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, et al (2017b) Mastering the game of go without human knowledge. *Nature* 550(7676):354.

Stone CJ (1982) Optimal global rates of convergence for nonparametric regression. *Ann. Statist.* 10(4):1040–1053.

Strehl AL, Li L, Wiewiora E, Langford J, Littman ML (2006) Pac model-free reinforcement learning. *Proc. 23rd Internat. Conf. Machine Learn.* (ACM, New York), 881–888.

Sturtevant NR (2008) An analysis of uct in multi-player games. van den Herik HJ, Xu X, Ma Z, Winands MHM, eds. *Computers and Games* (Springer, Berlin), 37–49.

Sutton RS (1988) Learning to predict by the methods of temporal differences. *Machine Learn.* 3(1):9–44.

Teraoka K, Hatano K, Takimoto E (2014) Efficient sampling method for Monte Carlo tree search problem. *IEICE Trans. Inform. Systems* 97(3):392–398.

Tsybakov AB (2009) *Introduction to Nonparametric Estimation*. Springer Series in Statistics (Springer, Berlin).

Van Hasselt H, Guez A, Silver D (2016) *Deep Reinforcement Learning with Double q-Learning*, vol. 2 (AAAI, Palo Alto, CA).

Watkins CJ, Dayan P (1992) Q-learning. *Machine Learn.* 8(3-4): 279–292.

Yang Y, Zhang G, Xu Z, Katabi D (2019) Harnessing structures for value-based planning and reinforcement learning. Preprint, version 3, submitted July 4, 2020, https://arxiv.org/abs/1909.12255.

Devavrat Shah is the Andrew and Erna Viterbi professor of electrical engineering and computer science at Massachusetts Institute of Technology. His research focuses on statistical inference and stochastic networks. His contributions span a variety of areas including resource allocation in communications networks, inference and learning on graphical models, algorithms for social data processing. He received the Erlang Prize and SIGMETRICS Rising Star Award.

Qiaomin Xie is an assistant professor in the Department of Industrial and Systems Engineering at the University of Wisconsin-Madison. Her research interests lie in the fields of reinforcement learning, applied probability, and stochastic networks.

Zhi Xu is affiliated with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. His research interests include both theoretical and applied machine learning.