Journal of Artificial Intelligence Research 51 (2014) 165-205 Submitted 05/14; published 09/14

Simple Regret Optimization in Online Planning for Markov Decision Processes

Zohar Feldman ZOHARF@TX.TECHNION.AC.IL
Carmel Domshlak DCARMEL@IE.TECHNION.AC.IL
Faculty of Industrial Engineering & Management,
Technion - Israel Institute of Technology,
Haifa, Israel

Abstract

We consider online planning in Markov decision processes (MDPs). In online planning, the agent focuses on its current state only, deliberates about the set of possible policies from that state onwards and, when interrupted, uses the outcome of that exploratory deliberation to choose what action to perform next. Formally, the performance of algorithms for online planning is assessed in terms of *simple regret*, the agent's expected performance loss when the chosen action, rather than an optimal one, is followed.

To date, state-of-the-art algorithms for online planning in general MDPs are either best effort, or guarantee only polynomial-rate reduction of simple regret over time. Here we introduce a new Monte-Carlo tree search algorithm, BRUE, that guarantees *exponential-rate* and *smooth* reduction of simple regret. At a high level, BRUE is based on a simple yet non-standard state-space sampling scheme, MCTS2e, in which different parts of each sample are dedicated to different exploratory objectives. We further extend BRUE with a variant of "learning by forgetting." The resulting parametrized algorithm, BRUE(α), exhibits even more attractive formal guarantees than BRUE. Our empirical evaluation shows that both BRUE and its generalization, BRUE(α), are also very effective in practice and compare favorably to the state-of-the-art.

1. Introduction

Markov decision processes (MDPs) offer a very general framework for sequential decision making under uncertainty (Puterman, 1994). An MDP $\langle S, A, Tr, R\rangle$ is defined by a set of possible agent states S, a set of agent actions A, a stochastic transition function $Tr : S\times A\times S \to [0, 1]$ defined by a set of $|S|\times|A|$ conditional probability functions $\mathcal{P}(S \mid s, a)$, and a reward function $R : S \times A \times S \to \mathbb{R}$. The current state of the agent is fully observable. When the agent performs action a at state s, the state changes to s' with probability $\mathcal{P}(s' \mid s, a)$, and the agent then collects a reward $R(s, a, s')$. In the finite horizon setting, the reward is accumulated over some predefined number of steps H.

The objective of the agent is to act so to maximize its accumulated reward, and the decision problem is always what action to perform next. For a state s, with h steps to go, a (possibly stochastic) action policy π prescribes an action to be taken in this situation. A policy is called optimal if, in expectation, following it guarantees maximization of the accumulated reward. The key property of the MDP model is that, for any MDP, there is a deterministic optimal policy $\pi^* : S \times \{1, \dots, H\} \to A$ (Bellman, 1957).

Efficiency of finding optimal policies for MDPs is the primary focus of the computational research around this model. When the state space of the MDP is too large for the allowed planning time, reasoning about the MDP is narrowed to a state space region that is considered most relevant to the specific decision problem currently faced by the agent. In particular, algorithms for *online* reasoning about MDPs focus only on the current state s_0 of the agent, deliberate about the set of possible courses of action from s_0 onwards, and, when interrupted, use the outcome of that exploratory deliberation, or planning, to issue an instant recommendation of an action to perform at s_0. Once that action is applied in the real environment, the planning process is repeated from the obtained state to select the next action and so on.

Depending on the problem domain and the representation language, concise descriptions of large-scale MDPs can be either declarative or generative (or mixed). With declarative representations, both transition and reward functions are described explicitly, while with generative models, they are given by a "black box" simulator. While the palette of algorithms for finding good actions in concisely represented MDPs is already rather wide (Boutilier, Dean, & Hanks, 1999; Guestrin, Koller, Parr, & Venkataraman, 2003; Kolobov, Mausam, & Weld, 2012; Busoniu & Munos, 2012; Bonet & Geffner, 2012; Keller & Helmert, 2013; Mausam & Kolobov, 2012; Geffner & Bonet, 2013), most of these algorithms are applicable only to declaratively represented MDPs. One of the earliest and best-known online planning algorithms developed for generative MDP models is the sparse sampling algorithm by Kearns, Mansour, and Ng (2002). Sparse sampling offers a near-optimal action selection in discounted MDPs by constructing a sampled lookahead tree in time exponential in the discount factor and sub-optimality bound, but independent of the state space size. However, if terminated before an action has proven to be near-optimal, sparse sampling offers no quality guarantees on its action selection.

In the last decade, *Monte-Carlo tree search (MCTS)* algorithms (Browne, Powley, Whitehouse, Lucas, Cowling, Rohlfshagen, Tavener, Perez, Samothrakis, & Colton, 2012) became extremely popular in online planning for MDPs, as well as in online planning for many other settings of sequential decision making, including those with partial state observability and adversarial effects (Gelly & Silver, 2011; Sturtevant, 2008; Bjarnason, Fern, & Tadepalli, 2009; Balla & Fern, 2009; Eyerich, Keller, & Helmert, 2010; Browne et al., 2012). The capability of dealing with generative problem representations was not the only feature of MCTS that made these methods so popular. First, while MCTS algorithms can natively exploit problem-specific heuristic functions, their correctness is independent of the heuristic's properties, and they can as well be applied without any heuristic information whatsoever. Second, numerous MCTS algorithms exhibit strong anytimeness: not only can a meaningful action recommendation be provided at any interruption point instantly, in time $O(1)$, but the quality of the recommendation is also improved very smoothly, in time steps that are independent of the size of the explored state space.

Formally, denoting by $s\langle h\rangle$ the state s with h steps-to-go, the quality of the action a, recommended for $s_0\langle H\rangle$, is assessed in terms of the *choice-error probability*, that is, the probability that a is sub-optimal, and in terms of the (closely related) measure of *simple regret* $\Delta[s\langle h\rangle, a]$. The latter captures the performance loss that results from taking a and then following an optimal policy π^* for the remaining $h-1$ steps, instead of following π^*

from the beginning (Bubeck & Munos, 2010).[1] That is,

$$\Delta[s\langle h\rangle, a] = Q(s\langle h\rangle, \pi^*(s\langle h\rangle)) - Q(s\langle h\rangle, a),$$

where

$$Q(s\langle h\rangle, a) = \begin{cases} \mathbb{E}_{s'}\left[R(s,a,s') + Q\left(s'\langle h{-}1\rangle, \pi^*(s'\langle h{-}1\rangle)\right)\right], & h > 0, \\ 0, & h = 0 \end{cases}.$$

Numerous MCTS algorithms, and in particular, the popular UCT (Kocsis & Szepesvári, 2006) algorithm and its variants (Coquelin & Munos, 2007; Tolpin & Shimony, 2012), guarantee eventual convergence to the optimal choice of action, while providing smooth reduction of the choice-error probability and simple regret over planning time. The relative empirical attractiveness of the various MCTS planning algorithms depends on the specifics of the problem at hand and usually cannot be predicted ahead of time. However, when it comes to formal guarantees on the expected performance improvement over the planning time, none of the online MCTS algorithms for MDPs breaks the barrier of the worst-case *polynomial-rate reduction* of simple regret and choice-error probability over time.

This is precisely our contribution here. Our work has been motivated by a recently growing understanding that the current MCTS algorithms for MDPs do not optimize the reduction of simple regret directly, but only via optimizing what is called cumulative regret, a performance measure suitable for the (very different) setting of "reinforcement learning while acting" (Bubeck & Munos, 2010; Busoniu & Munos, 2012; Tolpin & Shimony, 2012; Feldman & Domshlak, 2012).

- Departing from this high-level realization, we discuss certain pitfalls in simple regret minimization via Monte-Carlo sampling, and identify two, somewhat competing, exploratory objectives that should be pursued by the sampling mechanism. We then suggest a principle of "separation of concerns," whereby different parts of each state-space sample should be devoted to different exploration objectives.

- We introduce MCTS2e, a novel sampling scheme that specializes MCTS and implements that principle of "separation of concerns." Our main result is in the introduction and analysis of BRUE, a concrete instance of MCTS2e that guarantees *smooth* and *exponential-rate* reduction of both the simple regret and the choice-error probability over time, and this for general MDPs over finite state spaces. In fact, we show that qualitatively similar guarantees are satisfied by a broad class of what we call *purely exploring* MCTS2e algorithms, with BRUE being a simple yet efficient instance of this class.

- Finally, we discuss and analyze the prospects of "learning by forgetting," a principle according to which old samples are degraded as newer (and higher-quality) samples are gathered. Generalizing BRUE by extending it with this ingredient forms a

1. It may appear to the reader as more intuitive to consider the loss that results from applying the recommendations instead of π^* at all the H steps, and not just at the first one. However, as Kearns et al. (2002) show in their Lemma 5, the two measures are closely related, with this alternative measure being directly from simple regret along the execution horizon.

parametrized algorithm BRUE(α), with the parameter α controlling the level of "forgetfulness." We show that BRUE(α) exhibits even more attractive formal guarantees than those exhibited by BRUE.

The rest of the paper is structured as follows. In Section 2.2 we provide background on Monte-Carlo tree search, and in particular, on the UCT algorithm. Then, in Section 3, we discuss simple regret minimization in MDPs via a multi-armed bandits perspective, and in Section 4, we introduce the principle of "separation of concerns" and establish our main algorithmic constructs along with the corresponding computational results. Section 5 is devoted to "learning with forgetting" in MCTS, and in particular, to the algorithm BRUE(α). In Section 6 we discuss some findings of our empirical evaluation. The proofs of the formal claims are relegated to Appendix B, the three subsections of which contain, respectively, the proofs for the three key theorems. For completeness, Appendix A provides some standard concentration inequalities that we use in the paper.

2. Background

Henceforth, $A(s) \subseteq A$ denotes the actions applicable in state s, the operation of drawing a sample from a distribution $\mathcal{D}$ over set $\aleph$ is denoted by $\sim \mathcal{D}[\aleph]$, $\mathcal{U}$ denotes uniform distribution, and $[\![n]\!]$ for $n \in \mathbb{N}$ denotes the set $\{1, \ldots, n\}$. For a sequence of tuples ρ, $\rho[i]$ denotes the i-th tuple along ρ, and $\rho[i].x$ denotes the value of the field x in that tuple. When considering an MDP $\langle S, A, Tr, R\rangle$, K denotes the state branching factor (maximal number of actions per state), B denotes the action branching factor (maximal number of outcomes per action), and $\Delta = \min_{a \neq \pi^*(s_0,H)} \Delta[s_0\langle H\rangle, a]$ denotes the minimal possible non-zero simple regret at the root.

2.1 Sparse Sampling

One of the earliest and best-known online planning algorithms developed for generative MDP models is the sparse sampling (SS) algorithm by Kearns et al. (2002). While SS has been originally developed for infinite-horizon discounted MDPs, its reformulation for finite horizon MDPs is straightforward as follows.

For each action $a \in A(s_0)$, SS estimates its value $Q(s_0\langle H\rangle, a)$ by averaging C *recursive* samples of a's outcome states. The outcome states s' are sampled from the generative model of the transition function $Tr(s_0, a)$, and the value of such a sample is set to $R(s_0, a, s') + \max_{a'} Q(s'\langle H-1\rangle, a')$, with the Q-values of the actions $a' \in A(s')$ being estimated recursively the same way, until hitting depth H. The number of outcome samples C is set so to guarantee that the quality of the recommendation issued upon termination of the algorithm meets a desired level of accuracy. Alternatively, given C, the same formal analysis can be used to derive the corresponding accuracy guarantee. Equivalent bounds on simple regret are as follows.

Proposition 2.1.1 *Let* SS *be called on a state s_0 of an MDP $\langle S, A, Tr, R\rangle$ with rewards in $[0,1]$ and finite horizon H. Then, the simple regret of the action $\pi^{SS}(s_0\langle H\rangle)$, recommended by* SS *with parameter $C > 0$, is bounded as*

$$\mathbb{E}\Delta[s_0\langle H\rangle, \pi^{SS}(s_0\langle H\rangle)] \leq H(K \cdot \min(B,C))^H e^{-\frac{\Delta^2 C}{H^4}}.$$

The proof of Proposition 2.1.1 is given in Appendix C, p. 201. The bound in Proposition 2.1.1 suggests that the formal guarantees of SS become meaningful only when

$$C > \frac{H^5 \log(K \cdot \min\{B, C\})}{\Delta^2}.$$

Assuming that $B < C$, this implies that the bound in Proposition 2.1.1 becomes non-trivial only when the overall number of SS calls to the generative model is

$$O\left(\Delta^{-2} H^5 \log(BK)(BK)^H\right). \tag{1}$$

Notably, SS is not a strong anytime algorithm, but what is called a "contract" algorithm (Zilberstein, 1993): The termination of SS is parametrized by C, and interrupting SS before its normal termination results in no meaningful action recommendation. However, knowing the overall number of allowed calls to the generative model can in principle enable more knowledgeable allocation of the deliberation efforts (Hay, Shimony, Tolpin, & Russell, 2012). Hence, in general, deliverables of the contract algorithms are expected to be better than deliverables of the, *de facto* similarly budgeted, strong anytime algorithms (Zilberstein, 1993). Therefore, the bound in (1) sets a good reference for understanding the significance of formal guarantees provided by strong anytime algorithms for online MDP planning.

2.2 Monte-Carlo Tree Search and UCT

MCTS, a canonical scheme for Monte-Carlo tree search that gives rise to various specific algorithms for online MDP planning, is depicted in Figure 1, on the left. MCTS explores the state space in the radius of H steps from the initial state s_0 by iteratively rolling out state-space samples from s_0. Each such rollout ρ comprises a sequence of simulated steps $\langle s, a, s', r\rangle$ where s is a state, a is an action applicable in s, s' is a state resulting from applying a at s, and r is the corresponding immediate reward. In particular, $\rho[0].s = s_0$ and $\rho[t].s' = \rho[t{+}1].s$ for all t.

Each generated rollout is used to update some variables of interest associated with the states visited and actions applied therein. These variables typically include at least the action value estimators $\widehat{Q}(s\langle h\rangle, a)$, as well as the counters $n(s\langle h\rangle, a)$ that keep the number of times the corresponding estimators $\widehat{Q}\left(s\langle h\rangle, a\right)$ have been updated. The rollout-oriented exploration of MCTS allows information from states at deeper levels to be propagated to the root $s_0\langle H\rangle$ in low-complexity iterations of $O(H)$. This allows smooth improvement of the intermediate quality of recommendation, which is probably one of the main reasons that MCTS seems particularly appealing in the context of online planning.

Instances of MCTS vary mostly along the different implementation of the strategies

- StopRollout, specifying when to stop a rollout;
- RolloutAction, prescribing an action to apply in the current state of the rollout; and
- Update, specifying how a rollout should expand the tree $\mathcal{T}$ and update the maintained variables stored at the nodes of the constructed search tree.[2]

2. Due to the Markovian nature of MDPs, it is unreasonable to distinguish between nodes associated with the same state at the same depth. Hence, the actual graph constructed by most instances of MCTS forms a directed acyclic graph over nodes $s\langle h\rangle \in S \times \{0, 1, \ldots, H\}$.

```
MCTS: [input: ⟨S, A, Tr, R⟩; s0 ∈ S]

while time permits do
    ρ ← Rollout        // generate rollout
    Update(ρ)
return arg max_a Q̂(s0⟨H⟩, a)

procedure Rollout
    ρ ← ⟨⟩
    s ← s0
    d ← 0
    while not StopRollout(ρ) do
        h ← H − d
        a ← RolloutAction(s⟨h⟩)
        s′ ← RolloutOutcome(s⟨h⟩, a)
        r ← R(s, a, s′)
        ρ[t] ← ⟨s, a, r, s′⟩
        s ← s′; d ← d + 1
    return ρ
```

```
procedure Update(ρ)
    r̄ ← 0
    for d ← |ρ|, ..., 1 do
        h ← H − d
        a ← ρ[d].a
        n(s⟨h⟩) ← n(s⟨h⟩) + 1
        n(s⟨h⟩, a) ← n(s⟨h⟩, a) + 1
        r̄ ← r̄ + ρ[d].r
        MC-backup(s⟨h⟩, a, r̄)

procedure MC-backup(s⟨h⟩, a, r̄)
    Q̂(s⟨h⟩, a) ← ((n(s⟨h⟩, a) − 1) / n(s⟨h⟩, a)) Q̂(s⟨h⟩, a) + (1 / n(s⟨h⟩, a)) r̄

procedure StopRollout(ρ)
    d ← |ρ|
    return d = H or A(ρ[d].s′) = ∅

procedure RolloutAction(s⟨h⟩)        // UCB
    if ∃a : n(s⟨h⟩, a) = 0 then
        return a
    return argmax_a [ Q̂(s⟨h⟩, a) + c √( log n(s⟨h⟩) / n(s⟨h⟩, a) ) ]

procedure RolloutOutcome(s⟨h⟩, a)
    return s′ ∼ P(S | s, a)
```

Figure 1: A template for MCTS algorithms (left) , and the UCT algorithm as a specific set of sub-routines for MCTS (right).

Once interrupted, MCTS uses the information collected throughout the exploration to recommend an action to perform at state s_0.

Numerous concrete instances of MCTS have been proposed, with the UCT algorithm (Kocsis & Szepesvári, 2006) and its modifications (Coquelin & Munos, 2007; Tolpin & Shimony, 2011) being the most popular such instances these days (Gelly & Silver, 2011; Sturtevant, 2008; Bjarnason et al., 2009; Balla & Fern, 2009; Eyerich et al., 2010; Keller & Eyerich, 2012). The specification of the UCT algorithm as an instance of MCTS is depicted in Figure 1, on the right.

- Different versions of UCT use different rules to end a rollout. In the version depicted here, rollouts end at terminal nodes, that is, either at depth H or at states with no applicable actions.[3]

- The RolloutAction policy of UCT is based on the deterministic decision rule UCB1 (Auer, Cesa-Bianchi, & Fischer, 2002), originally proposed for optimal balance between exploration and exploitation for cumulative regret minimization in stochastic

3. In a more popular version of UCT, the search tree is grown incrementally, by ending the rollouts whenever a new node is encountered. However, this point is extraneous for the exposition of this paper.

multi-armed bandit (MAB) problems (Robbins, 1952). At node $s\langle h\rangle$, the next-on-the-sample action a is selected as follows: If all the actions applicable in s have been sampled by now at $s\langle h\rangle$, that is, if $n(s\langle h\rangle, a) > 0$ for all $a \in A(s)$, then the selected action corresponds to

$$\underset{a}{\operatorname{argmax}} \left[\widehat{Q}(s\langle h\rangle, a) + c\sqrt{\frac{\log n(s\langle h\rangle)}{n(s\langle h\rangle, a)}} \right], \tag{2}$$

where $c > 0$ is a fixed parameter that balances between the first, exploitation-oriented, and the second, exploration-oriented, summands in Eq. 2. Otherwise, a is selected uniformly at random from the still unexplored actions $\{a \in A(s) \mid n(s\langle h\rangle, a) = 0\}$. In both cases, the procedure SAMPLE-OUTCOME of UCT then samples the next state on the rollout according to the transition probability $\mathcal{P}(S \mid s, a)$.

- UCT updates all the value estimators $\widehat{Q}(s\langle h\rangle, a)$ of the $(s\langle h\rangle, a)$ pairs encountered along the rollouts. The updates are done via the MC-BACKUP procedure, which averages the accumulated rewards of the rollouts from $s\langle h\rangle$ to terminal states.

In terms of formal properties, UCT is an online algorithm that, from a certain point in time, provides a smooth reduction of simple regret over time to zero, that is, a smooth convergence to the optimal action choice at $s_0\langle H\rangle$ (Kocsis & Szepesvári, 2006). Two aspects of convergence are of interest: (1) the *length of the transition period* during which no reduction of simple regret can be guaranteed at all, and (2) the *reduction rate* of simple regret over time, after the transition period is over. Considering (1), Coquelin and Munos (2007) showed that the number of samples after which the bounds of UCT on simple regret become meaningful might be as high as *hyper-exponential in H*. Considering (2), Theorem 6 in the work of Kocsis and Szepesvári (2006) claims a *polynomial-rate reduction* of the probability of choosing a non-optimal action, which implies the same for the simple regret[4].

Some attempts have recently been made to improve UCT, and online MCTS-based planning in general, in terms of these two aspects of convergence (Tolpin & Shimony, 2012; Hay et al., 2012; Coquelin & Munos, 2007). While the reported empirical results were promising, none of these suggested MCTS instances breaks the UCT's barrier of the worst-case polynomial-rate reduction of simple regret over time. Hence, the question of *whether an online, smoothly converging MCTS algorithm can substantially outperform* UCT *in terms of these two convergence parameters* remained open. In what comes next, we answer this question affirmatively.

3. Simple Regret Minimization in MDPs

At a high level, the key property of UCT is that its exploration of the search space is obtained by considering a hierarchy of forecasters (s, h), each minimizing its own *cumulative regret*, that is, the loss of the total reward incurred while exploring the environment (Auer et al., 2002). In that respect, according to Theorem 6 in the work of Kocsis and Szepesvári (2006), UCT asymptotically achieves the best possible (logarithmic) cumulative regret. However, as recently pointed out in numerous works (Bubeck & Munos, 2010; Busoniu & Munos, 2012;

4. Notably, these claims are made under some nontrivial assumptions

Tolpin & Shimony, 2012; Feldman & Domshlak, 2012), cumulative regret does not seem to be the right objective for online MDP planning, and this is because the rewards "collected" at the simulation phase are fictitious. Furthermore, the work of Bubeck, Munos, and Stoltz (2011) on multi-armed bandits shows that minimizing both the cumulative and the simple regret are somewhat competing objectives, in the sense that the minimal simple regret can increase as the bound on the cumulative regret decreases.

This relationship between simple and cumulative regret minimization in MABs suggests that focusing online MDP planning directly on simple regret minimization may lead to algorithms that are, worst-case and/or empirically, substantially more effective than UCT. In fact, in the context of MABs, Bubeck et al. (2011) already showed that a simple round-robin sampling of MAB actions, followed by recommending the action with the highest empirical mean, yields exponential-rate reduction of simple regret, while the UCB1 sampling strategy that balances between exploration and exploitation yields only polynomial-rate reduction of that measure. In that respect, the situation with MDPs is seemingly no different. In fact, although designed for a slightly different setup, the sparse sampling algorithm provides an evidence for the theoretical merits of being focused solely on exploration in online planning for MDPs.

It appears, however, that the answer to the question of how one should "focus on exploration," while preserving *both onlineness and smoothness of convergence*, is less straightforward in general MDPs than it is in the special case of MABs. Before we motivate and discuss the various exploratory concerns in online Monte-Carlo planning for MDPs, and what the separation of these concerns can possibly buy us, we begin with a MAB perspective on MDPs, which shows that smooth exponential-rate reduction of simple regret in MDPs is indeed achievable, at least theoretically.

3.1 Multi-armed Bandit Perspective on MDPs

Let s_0 be a state of an MDP $\langle S, A, Tr, R\rangle$ with rewards in $[0, 1]$, and a finite horizon H. In principle, such a general MDP can be viewed as a MAB, with each arm in the MAB corresponding to a "flat" policy of acting for H steps starting from the current state s_0. A "flat" policy π is a minimal partial mapping from state/steps-to-go pairs to actions that fully specifies an acting strategy in the MDP for H steps, starting at s_0. Sampling such an arm π is straightforward as π prescribes precisely which action should be applied at every state that can possibly be encountered along the execution of π. The reward of such an arm π is stochastic, with support $[0, H]$, and expected value μ_π. The number of arms in this schematic MAB is $K^{\sum_{i=0}^{H-1} B^i} \approx K^{B^H}$. Now, consider a simple algorithm, NaiveUniform, which systematically samples each "flat" policy in a loop, and uses the obtained reward to update the empirical mean $\widehat{\mu}_\pi$ of the corresponding policy arm π. If stopped at iteration n, the algorithm recommends the policy arm π_n with the best empirical value $\widehat{\mu}_{\pi_n}$. By iteration n of this algorithm, each arm will be sampled at least $\lfloor \frac{n}{K^{B^H}} \rfloor$ times. Therefore, using Hoeffding's tail inequality[5], the probability that the chosen arm policy π_n is sub-

5. For completeness, Hoeffding's tail inequality is provided in Appendix A, pp. 188.

optimal in our MAB is upper-bounded by

$$\sum_{\pi \neq \pi^*} \mathbb{P}\left\{\widehat{\mu}_\pi > \widehat{\mu}_{\pi^*}\right\} = \sum_{\pi \neq \pi^*} \mathbb{P}\left\{\widehat{\mu}_\pi - \widehat{\mu}_{\pi^*} - (-\Delta_\pi) \geq \Delta_\pi\right\} \leq K^{B^H} e^{-\frac{\lfloor \frac{n}{K^{B^H}} \rfloor \Delta^2}{2H^2}}, \tag{3}$$

where $\Delta_\pi = \mu_{\pi^*} - \mu_\pi$ and $\Delta = \min_{\pi \neq \pi^*} \Delta_\pi$. Denoting the simple regret of π_n by r_n , the expected simple regret can therefore be bounded as

$$\mathbb{E} r_n \leq H K^{B^H} e^{-\frac{\lfloor \frac{n}{K^{B^H}} \rfloor \Delta^2}{2H^2}}. \tag{4}$$

Note that NaiveUniform uses each rollout $\rho = \langle s_0\langle H\rangle, a_1, s_1\langle H-1\rangle, \ldots, a_H, s_H\langle 0\rangle\rangle$ to update the estimation of only a single policy π. However, recalling that arms in our MAB problem are actually compound policies, the same sample can in principle be used to update the estimates of all policies π' that are consistent with ρ in the sense that, for $0 \leq i \leq H-1$, $\pi'(s_i\langle H-i\rangle)$ is defined and $\pi'(s_i\langle H-i\rangle) = a_{i+1}$. The resulting algorithm, CraftyUniform, generates samples by choosing the actions along the sample uniformly at random, and uses the outcome of each sample to update all the policies consistent with it. Note that the policy arms in CraftyUniform cannot be sampled systematically as in NaiveUniform because the set of policies updated at each iteration is stochastic.

Since the sampling is uniform, the probability of any policy to be updated by a sample issued at any iteration of CraftyUniform is $\frac{1}{K^H}$. Let $N_\pi \leq n$ denote the number of samples consistent with the policy π among the first n samples issued by CraftyUniform. The probability that π_n, the best empirical arm after n iterations, is sub-optimal is bounded by $\sum_{\pi \neq \pi^*} \mathbb{P}\left\{\widehat{\mu}_\pi > \widehat{\mu}_{\pi*}\right\}$ where

$$\mathbb{P}\left\{\widehat{\mu}_\pi > \widehat{\mu}_{\pi*}\right\} \leq \mathbb{P}\left\{\widehat{\mu}_\pi - \mu_\pi \geq \frac{\Delta_\pi}{2}\right\} + \mathbb{P}\left\{\widehat{\mu}_{\pi*} - \mu_{\pi*} \geq \frac{\Delta_\pi}{2}\right\}. \tag{5}$$

Each of the two terms on the right-hand side can be bounded as:

$$\begin{aligned}
\mathbb{P}\left\{\widehat{\mu}_\pi - \mu_\pi \geq \frac{\Delta_\pi}{2}\right\} &\leq \mathbb{P}\left\{N_\pi \leq \frac{n}{2K^H}\right\} + \mathbb{P}\left\{N_\pi > \frac{n}{2K^H}, \, \widehat{\mu}_\pi - \mu_\pi \geq \frac{\Delta_\pi}{2}\right\} \\
&\overset{(\dagger)}{\leq} e^{-\frac{n}{8K^H}} + \sum_{i=\frac{n}{2K^H}+1}^{n} \mathbb{P}\left\{N_\pi = i\right\} \mathbb{P}\left\{\widehat{\mu}_\pi - \mu_\pi \geq \frac{\Delta_\pi}{2} \,\middle|\, N_\pi = i\right\} \\
&\leq e^{-\frac{n}{8K^H}} + \mathbb{P}\left\{\widehat{\mu}_\pi - \mu_\pi \geq \frac{\Delta_\pi}{2} \,\middle|\, N_\pi = \frac{n}{2K^H} + 1\right\} \sum_{i=\frac{n}{2K^H}+1}^{n} \mathbb{P}\left\{N_\pi = i\right\} \\
&\leq e^{-\frac{n}{8K^H}} + \mathbb{P}\left\{\widehat{\mu}_{\pi,n} - \mu_\pi \geq \frac{\Delta_\pi}{2} \,\middle|\, N_\pi = \frac{n}{2K^H} + 1\right\} \\
&\overset{(\ddagger)}{\leq} e^{-\frac{n}{8K^H}} + e^{-\frac{n\Delta_\pi^2}{4K^H H^2}} \\
&\leq 2e^{-\frac{n\Delta_\pi^2}{8K^H H^2}},
\end{aligned} \tag{6}$$

where (†) and (‡) are by Hoeffding's tail inequality. In turn, similarly to Eq. 4, the simple regret for CraftyUniform is bounded by

$$\mathbb{E}r_n \leq 4HK^{B^H}e^{-\frac{n\Delta^2}{8K^H H^2}}. \tag{7}$$

Since H is a trivial upper-bound on $\mathbb{E}r_n$, the bound in Eq. 7 becomes effective only when $4K^{B^H}e^{-\frac{n\Delta^2}{8K^H H^2}} < 1$, that is, for

$$n > (KB)^H \cdot 4\left(\frac{H}{\Delta}\right)^2 \log K. \tag{8}$$

Note that this "cold start" transition period is *much* shorter than that of UCT, which can be hyper-exponential in H. At the same time, unlike in UCT, the rate of the simple regret reduction here is exponential in the number of iterations. In terms of oracle calls, the length of the transition period for CraftyUniform is

$$O\left(\Delta^{-2}H^3\log(K)\,(KB)^H\right).$$

Likewise, comparing to sparse sampling (Eq. 1), it appears that the transition period of CraftyUniform has a smaller dependency on H (H^3 vs. H^5), and a smaller dependency on B ($\log(K)$ vs. $\log(BK)$).

In sum, CraftyUniform can be seen as a theoretical feasibility test for our agenda: The algorithm uses Monte-Carlo sampling and averaging updates, it is strong anytime (action recommendation can be issued instantly, at any time, and the expected quality of the recommendation improves after every state-space sample), and simple regret decreases at an exponential rate over time. Moreover, the transition period after which this reduction rate is guaranteed is somewhat shorter than the (contracted) transition period of SS, and it is much shorter than the transition period of UCT. In any case, however, the feasibility of CraftyUniform is only conceptual: it requires explicit reasoning about K^{B^H} arms, and thus it cannot be efficiently implemented.

4. Separation of Concerns in Online MDP Planning

We now show a practical algorithm that achieves smooth, exponential-rate reduction of simple regret in online MDP planning. To do so, we first motivate and introduce a principle of "separation of concerns," whereby different parts of each state-space sample are devoted to different aspects of problem exploration. We then introduce MCTS2e, a specialized MCTS sampling scheme that implements that principle of "separation of concerns" via a two-phase scheme for generating state-space samples. Using MCTS2e as our basis, we describe a concrete algorithm, BRUE, that achieves exponential-rate, smooth reduction of simple regret over time, and has a transition period comparable to these of the schematic CraftyUniform and of the non-interruptible SS. In fact, we show that these formal guarantees are satisfied by the entire class of what we call "purely exploring" MCTS2e algorithms, one of which is BRUE.

If we tried to achieve smooth, exponential-rate convergence by merely replacing the UCB1 policy of UCT with a "pure exploration" policy such as uniform action selection, then we would have failed miserably. In fact, this naive attempt would result in an algorithm that does not even converge to the optimal action. The reason for that lies in a fundamental difference between MABs and MDPs: Unlike in MABs, direct sampling of the actual value of the actions is impossible because doing so requires knowledge of the optimal policy at subsequent states in the entire look-ahead space. This knowledge, however, is unavailable at the beginning of deliberation. Hence, when sampling the futures, each non-root node $s\langle h\rangle$ should actually serve *two* objectives:

(1) estimating the actions at the ancestor(s) of $s\langle h\rangle$ in $\mathcal{T}$, and

(2) identifying the optimal action $\pi^*(s\langle h\rangle)$.

While both these objectives are *exploratory*, they are in opposition to some extent. To meet the first objective, $s\langle h\rangle$ should sample its optimal action $\pi^*(s\langle h\rangle)$ with a probability approaching 1 as the number of samples grows. To meet the second objective, however, *all* actions at $s\langle h\rangle$ must be selected frequently. When the same protocol for selecting actions is used, as in UCT, throughout the entire rollout, and the rewards collected along this rollout are used for updating value estimations at multiple nodes, this protocol should commit to addressing these two objectives simultaneously. For instance, the UCB1 protocol employed by UCT at all nodes $s\langle h\rangle$ chooses the action that seems most attractive in potential, where this potential stems partially from the relatively high empirical value (complying with objective (1)), and partially from the less frequent sampling of that action (complying with objective (2)).

However, while such an overloading of the action selection protocol is unavoidable in the "learning while acting" setup of reinforcement learning, this is not the case in online planning. In some sense, the two objectives depicted above resemble the two tasks faced by MAB forecasters: objective (1) can be seen as a type of recommendation, whereas objective (2) can be viewed as exploration. It therefore makes perfect sense to fulfill these two objectives by different policies, much like exploration and recommendation are handled by different policies in MAB online planning (Bubeck et al., 2011). More specifically, different policies can be used to choose the method by which node/action pairs should be updated and the method by which the values of these pairs should be estimated. In what follows, we refer to this separation of exploratory objectives as "separation of concerns," and next we elaborate on the implementation of this concept in online planning for MDPs.

4.1 Two-Phase Sampling and BRUE

We now introduce a novel Monte-Carlo tree search scheme, MCTS2e, tailored towards employing the principle of "separation of concerns." MCTS2e is depicted in Figure 2(a) as a specification of MCTS's Update procedure. The core difference between the MCTS2e implementation of Update and that of UCT is in the samples used to update the value estimators. As illustrated in Figure 3, the value estimators in UCT are updated with the accumulated reward from the respective tail of the rollout, whereas in MCTS2e the estimators are updated with the accumulated reward of *new sub-rollouts*, created by the Estimate procedure.

The Estimate procedure is parametrized with two policies, namely

— EstAction, prescribing the action used for estimation, and

procedure Update(ρ)
  for $d \leftarrow |\rho|, \dots, 1$ **do**
    $h \leftarrow H - d$
    $\langle s, a, r, s' \rangle \leftarrow \rho[d]$
    $n(s\langle h \rangle) \leftarrow n(s\langle h \rangle) + 1$
    $n(s\langle h \rangle, a) \leftarrow n(s\langle h \rangle, a) + 1$
    $n(s\langle h \rangle, a, s') \leftarrow n(s\langle h \rangle, a, s') + 1$
    $\bar{r} \leftarrow r + \text{Estimate}(s'\langle h-1 \rangle)$
    MC-backup($s\langle h \rangle, a, \bar{r}$)

procedure Estimate($s\langle h \rangle$)
  $\bar{r} \leftarrow 0$
  for $d \leftarrow 0, \dots, h-1$ **do**
    $a \leftarrow \text{EstAction}(s\langle h-d \rangle)$
    $s' \leftarrow \text{EstOutcome}(s\langle h-d \rangle, a)$
    $\tilde{r}_{d+1} \leftarrow R(s, a, s')$
    $\bar{r} \leftarrow \bar{r} + \tilde{r}_{d+1}$
    $s \leftarrow s'$
  return $\bar{r}$

(a)

procedure StopRollout(ρ)
  $d \leftarrow |\rho|$
  return $d = H$ **or** $A(\rho[d].s') = \emptyset$

procedure RolloutAction($s\langle h \rangle$)   // *uniform*
  return $a \sim \mathcal{U}[A(s)]$

procedure RolloutOutcome($s\langle h \rangle, a$)
  return $s' \sim \mathcal{P}(S \mid s, a)$

procedure EstAction($s\langle h \rangle$)   // *best*
  return $\text{argmax}_{a \in A(s)} \widehat{Q}(s\langle h \rangle, a)$

procedure EstOutcome($s\langle h \rangle, a$)
  for $s' : n(s\langle h \rangle, a, s') > 0$ **do**
    $\widehat{\mathcal{P}}(S = s' \mid s, a) \leftarrow \frac{n(s\langle h \rangle, a, s')}{n(s\langle h \rangle, a)}$
  return $s' \sim \widehat{\mathcal{P}}(S \mid s, a)$

(b)

Figure 2: (a) MCTS2e as MCTS with specific Update procedure, and (b) the BRUE algorithm as a specific set of sub-routines for MCTS2e (right).

— EstOutcome, determining the next state to follow.

The policies RolloutAction and RolloutOutcome (used by the MCTS's Rollout procedure) determine what value estimators to update, while the policies EstAction and EstOutcome are used to update these estimators.

This separation allows us to introduce BRUE, which is, in a way, the most "exploratory" MCTS2e instance possible.[6] The BRUE setting of MCTS2e is depicted in Figure 2(b). Similarly to UCT, the rollouts generated in BRUE end at terminal nodes, and, throughout the rollout, the next state is sampled according to Tr. However, unlike in UCT, the rollout actions in BRUE are selected *uniformly at random* from all the applicable actions. In turn, in the estimation sub-rollouts,

— the selected actions are the empirically best actions, that is, the actions that have the highest value estimations, and

— the next states are sampled according to the empirical transition probabilities $\widehat{\mathcal{P}}(S = s' \mid s, a)$, that is, the number of times $n(s\langle h \rangle, a, s)$ state s' was followed when applying

6. Short for **B**est **R**ecommendation with **U**niform **E**xploration; the name is carried on from our first presentation of the algorithm, where "estimation" was referred to as "recommendation" (Feldman & Domshlak, 2012).

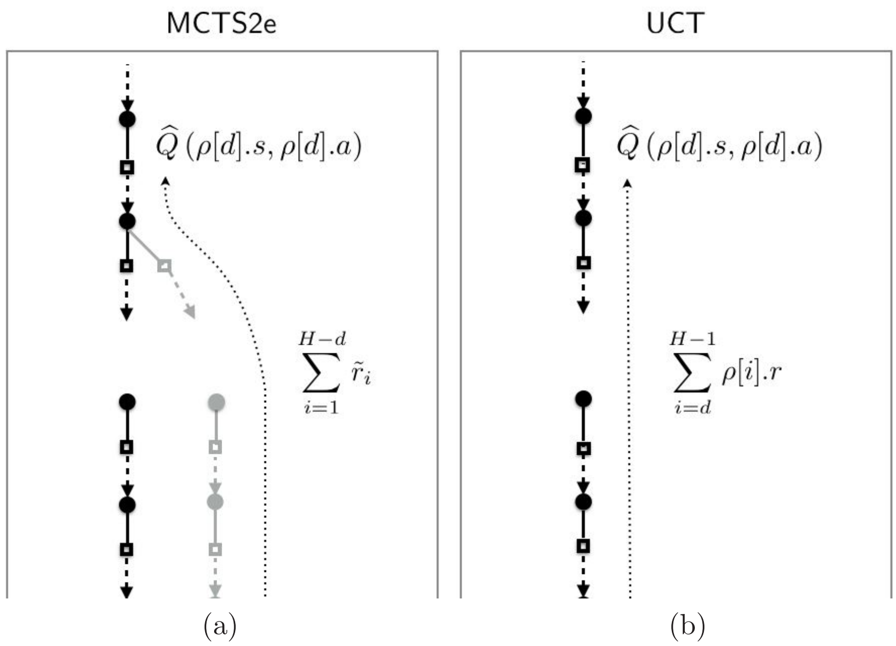


Figure 3: Illustration of a value estimator update in MCTS2e (a) vs. UCT (b). Circles represent decision nodes, solid lines represent the actions taken, squares represent the chance nodes, and dashed arrows represent the outcomes that result in the subsequent decision nodes.

action a in node $s\langle h\rangle$, divided by the overall number of times $n(s\langle h\rangle, a)$ that action a was applied in $s\langle h\rangle$.[7]

We now proceed with a formal analysis of BRUE. In general, when considering an instance of MCTS2e, by $\mathcal{T}_n$ we denote the search graph obtained after n iterations. For the sake of simplicity, we assume uniqueness of the optimal policy π^*: at each state s and each number h of steps-to-go, we assume a single optimal action, and denote it by $\pi^*(s, h)$. For all nodes $s\langle h\rangle \in \mathcal{T}_n$, $\pi_n^{\mathrm{B}}(s\langle h\rangle)$ is a randomized strategy, uniformly choosing among actions a maximizing $\widehat{Q}(s\langle h\rangle, a)$. In addition to the problem-specific state branching factor K, and minimal non-zero simple regret at the root $\Delta = \min_{a\neq\pi^*(s_0,H)} \Delta[s_0\langle H\rangle, a]$, our bounds below depend on the problem-specific action branching factor B, as well as on the horizon H. The former two parameters are inherited from MAB, while the latter two connect between MAB and general MDP.

Theorem 1 *Let* BRUE *be called on a state s_0 of an MDP $\langle S, A, Tr, R\rangle$ with rewards in $[0, 1]$, and finite horizon H. There exist pairs of parameters $\mathbf{a}, \mathbf{b} > 0$, dependent only on*

7. Sampling according to $\mathcal{P}(S \mid s, a)$ as in RolloutOutcome is also a valid choice, although in terms of formal guarantees, EstOutcome as in Figure 2 appears to be more attractive.

$\{K, B, H, \Delta\}$, such that, after $n > H$ iterations of BRUE*, the simple regret is bounded as*

$$\mathbb{E}\Delta[s_0\langle H\rangle, \pi_n^B(s_0\langle H\rangle)] \leq H\mathbf{a} \cdot e^{-\mathbf{b}n}, \tag{9}$$

and the choice-error probability is bounded as

$$\mathbb{P}\left\{\pi_n^B(s_0\langle H\rangle) \neq \pi^*(s_0\langle H\rangle)\right\} \leq \mathbf{a} \cdot e^{-\mathbf{b}n}. \tag{10}$$

In particular, Eq. 9 and 10 hold with $\mathbf{a} = 3K\left(\frac{1044B^2K^2}{\Delta^2}\right)^{H-1}(196BK)^{\frac{1}{2}(H-1)^2}(H-1)!^2$ and $\mathbf{b} = \frac{\Delta^2}{9K^2(196BK)^{H-1}H^2}$.

The proof of Theorem 1 is given in Appendix B.1, p. 189. The length of the transition period implied by Theorem 1 is given by

$$O\left(\Delta^{-2}H^5\log(\frac{BK}{\Delta})(196BK)^H\right) \tag{11}$$

This transition period is rather comparable to that of sparse sampling except for the rather large constant appearing in the basis of the exponent in Equations 9 and 10. Although this constant imposes a significant increase of the transition period, few things should be noted with regards to the bounds provided for BRUE. First and foremost, the parameter $\mathbf{b}$ in Theorem 1 reflects the worst-case in terms of the transition function Tr, which corresponds to a uniform distribution, that is, $\mathcal{P}(s' \mid s, a) = \frac{1}{B}$ for all states s and actions $a \in A(s)$. However, if the probability mass of the action transition functions each concentrates on a small set of outcomes, then the convergence rate of BRUE is expected to be much better. Proposition 4.1.1 formulates BRUE's bounds with respect to a problem-dependent parameter $1 \leq P_e \leq B$, which is related to the entropy of the transition function and is defined as

$$P_e = \max_{s,a} \|\mathcal{P}(\cdot \mid s, a)\|_{\frac{1}{2}}.$$

Proposition 4.1.1 *Let* BRUE *be called on a state s_0 of an MDP $\langle S, A, Tr, R\rangle$ with rewards in $[0, 1]$, and finite horizon $H \geq 4$. Then* BRUE *converges at an exponential rate in the sense of Eq. 9 and 10 of Theorem 1 with $\mathbf{a} = 3K\left(\frac{172BK}{\Delta^2}\right)^{H-1}(H!)^2$ and $\mathbf{b} = \frac{\Delta^2}{9KB^4(1666K)^{H-1}P_e^{H-5}H^2}$.*

The formal guarantees of BRUE can therefore be *even better* than these for SS. The proof for Proposition 4.1.1 (given in Appendix B.2, p. 195) is obtained by a rather minor modification of the proof for Theorem 1.

Relating to the tightness of the bounds, it should be noted that the size of the scalar constants in the analysis of BRUE partially stems from our attempt to avoid cumbersome expressions, and thus can be considerably reduced. Furthermore, in a particular point in our analysis where we bound the error of action-value estimations at different points in time, we believe that our bound gets particularly loose. We comment about this issue in more detail within the proof of Theorem 1 (right after Proposition B.1.1, p. 191).

Having read this far, the reader may rightfully ask to what extent the guarantees provided by BRUE are unique among the instances of MCTS2e. In general, the formal properties of MCTS2e instances heavily depend on their specific sub-routines, and some of them will not even guarantee convergence to the optimal action. However, BRUE is still very much not unique in its deliverables. In particular, below we define a family of *purely exploring* MCTS2e algorithms that all guarantee exponential-rate reduction of simple regret over time.[8]

Definition 1 (Purely exploring MCTS2e) *An instance $\mathcal{A}$ of* MCTS2e *is called* purely exploring *if, for each node $s\langle h\rangle$ reachable from s_0, and each $a \in A(s)$, there exist parameters ξ, β, γ, dependent only on $\{K, B, H, \Delta\}$, such that*

$$\mathbb{P}\left\{n(s\langle h\rangle, a) \leq \xi n(s\langle h\rangle)\right\} \leq \beta e^{-\gamma n(s\langle h\rangle)},$$

and the estimation policy EstAction *selects the empirically best arm.*

Theorem 2 *Let $\mathcal{A}$ be a purely exploring instance of* MCTS2e. *Then $\mathcal{A}$ converges at an exponential rate in the sense of Eq. 9 and 10 of Theorem 1.*

In Appendix B.3, p. 196 we show how a proof of Theorem 2 can be easily derived from our proof of Theorem 1. Furthermore, the analysis provided in the proof for Theorem 1 can be used to extract the convergence parameters c, c' for any purely exploring algorithm, given its specific parameters ξ, β, γ.

5. Learning With Forgetting and BRUE(α)

In BRUE, as well as in other converging instances of both MCTS and MCTS2e, the evolution of action value estimates at the internal nodes is based on biased samples that stem from the selection of non-optimal actions at the descendant nodes. This bias tends to shrink as more samples are accumulated at these descendants. Consequently, the estimates become more accurate, the probability of selecting an optimal action increases accordingly, and the bias of the ancestor nodes shrinks in turn.

An interesting question that arises in this context is whether samples obtained at different stages of the sampling process should be weighed differently. At a high level, our intuition suggests that biased samples do provide us with some valuable information, especially when they are still all we have. At the same time, the value of this information decreases as we obtain more accurate samples. Hence, in principle, putting more weight on samples with smaller bias could increase the accuracy of our estimates. This led us to consider BRUE(α), an algorithm that generalizes BRUE $\equiv$ BRUE(1) by basing the estimates only on the α *fraction of most recent samples*.

Technically, BRUE(α) differs from BRUE only in the implementation of the MC-backup procedure as depicted in Figure 4. In addition to the variables maintained by BRUE, each node/action pair $(s\langle h\rangle, a)$ in BRUE(α) is associated with a *list* $\mathcal{L}(s\langle h\rangle, a)$ of rewards, collected at each of the $n(s\langle h\rangle, a)$ samples that are responsible for the current estimate $\widehat{Q}(s\langle h\rangle, a)$. When $(s\langle h\rangle, a)$ is updated by MC-backup, the value estimator $\widehat{Q}(s\langle h\rangle, a)$ is assigned with the average of the most recent $\lceil \alpha \cdot n(s\langle h\rangle, a)\rceil$ samples, where $\lceil x\rceil$ denotes the

8. We, of course, make no claims that these guarantees are exclusive to the purely exploring instances of MCTS2e, or even to MCTS2e instances in general.

$$\begin{array}{l}
\textbf{procedure } \text{MC-BACKUP}(s\langle h\rangle, a, \bar{r}) \\
\quad n_\alpha \leftarrow \lceil \alpha \cdot n(s\langle h\rangle, a) \rceil \\
\quad n \leftarrow n(s\langle h\rangle, a) \\
\quad \mathcal{L}(s\langle h\rangle, a)[n] \leftarrow \bar{r} \\
\quad \widehat{Q}(s\langle h\rangle, a) \leftarrow \frac{1}{n_\alpha} \sum_{i=n-n_\alpha}^{n} \mathcal{L}(s\langle h\rangle, a)[i]
\end{array}$$

Figure 4: BRUE(α) modified MC-BACKUP procedure

smallest integer that is greater than or equal to x. Theorem 3 below exhibits the benefits of adopting $\alpha < 1$ when it comes to convergence guarantees.

Theorem 3 *Let* BRUE(α) *be called on a state* s_0 *of an MDP* $\langle S, A, Tr, R\rangle$ *with rewards in* $[0, 1]$ *and finite horizon* H. *There exist pairs of parameters* $\mathbf{a}, \mathbf{b} > 0$, *dependent only on* $\{K, B, H, \Delta, \alpha\}$, *such that, after* $n > H$ *iterations of* BRUE (α), *we have simple regret bounded as*

$$\mathbb{E}\Delta[s, \pi_n^B(s_0, H), H] \leq H\mathbf{a} \cdot e^{-\mathbf{b}n}, \tag{12}$$

and choice-error probability bounded as

$$\mathbb{P}\left\{\pi_n^B(s_0, H) \neq \pi^*(s_0, H)\right\} \leq \mathbf{a} \cdot e^{-\mathbf{b}n}. \tag{13}$$

In particular, for a depth-dependent $\alpha_h \approx \frac{1}{(BK)^{h-1}}$, *Eq. 12 and 13 hold with* $\mathbf{a} = 3K\left(\frac{12BK}{\Delta^2}\right)^{H-1}(H!)^2$ *and* $\mathbf{b} = \frac{\Delta^2}{9K^2(196BK)^{H-1}H^2}$.

For the particular choice of α_h in Theorem 3, the length of the transition period of BRUE(α) in terms of number of calls to the generative model is

$$O\left(\Delta^{-2}H^4 \log\left(\frac{BKH}{\Delta}\right)(196BK)^H\right).$$

While the bound for BRUE(α) seems somewhat better than that of BRUE, this improvement should be attributed more to the looseness of the bound for BRUE and less to the actual improvement in performance. The proof for Theorem 3 (given in Appendix B.4, p. 196) does not offer a new technique to address the bound on the accuracy of action-value estimations at different sampling times, but it reduces the bound by considering fewer samples. The selection of α in Theorem 3 stems from an attempt to balance as much as possible between the two sources of inaccuracy appearing in Propositions B.4.1 and B.4.2 of the proof for Theorem 3. The smaller α is, the lower is the sample inaccuracy that originates from the inaccuracy of the estimates at the successor nodes. At the same time, however, the inaccuracy that stems from basing the estimate on a fewer samples increases. Due to the branching, nodes farther toward the horizon are sampled less frequently and thus are less accurate. In the worst case, when the underlying graph $\mathcal{T}_n$ is a tree, a node is expected to be sampled only a fraction $\frac{1}{(BK)^d}$ of the number of samples that were taken at its "d steps higher" predecessor. This is precisely the reason for the selection of $\alpha_h \approx \frac{1}{(BK)^{h-1}}$ in Theorem 3.

In practice, however, these worst-case considerations tend to underrate the value of samples. Since $\mathcal{T}_n$ is typically not a tree, the ratio between the number of samples at different depths tends to be higher than the aforementioned worst-case ratio. Therefore, α should better be adapted according to the observed ratios rather than according to the worst-case ones. Furthermore, since our objective behind estimating the action values is to identify the optimal action, the bias of the samples may have far less influence on the quality of the planning outcome than that dictated by the formal guarantees. For instance, suppose that all action estimators at a particular node $s\langle h\rangle$ have an equal bias. If this is the case, then $s\langle h\rangle$ may home in on its optimal action while $\widehat{Q}(s\langle h\rangle, a)$ estimates are still biased, and that will suffice for $s\langle h\rangle$ to fulfill its role in value-estimating sub-rollouts issued by its ancestor(s). While this illustrative setup is clearly extreme, the point here is that biased estimators can still distinguish the better actions from the worse ones, as long as the biases across the actions are correlated.

6. Experimental Evaluation

We have evaluated BRUE empirically on the MDP Sailing domain (Péret & Garcia, 2004), used in previous works for evaluating MCTS algorithms (Péret & Garcia, 2004; Kocsis & Szepesvári, 2006; Tolpin & Shimony, 2012), as well as on an MDP version of random game trees used in the original empirical evaluation of UCT (Kocsis & Szepesvári, 2006).

In the Sailing domain, a sailboat navigates to a destination on an 8-connected grid representing a marine environment, under fluctuating wind conditions. The goal is to reach the destination as quickly as possible, by choosing at each grid location a neighbor location to move to. The duration of each such move depends on the direction of the move (*ceteris paribus*, diagonal moves take $\sqrt{2}$ more time than straight moves), the direction of the wind relative to the sailing direction (the sailboat cannot sail against the wind and moves fastest with a tail wind), and the tack. The direction of the wind changes over time, but its strength is assumed to be fixed. This sailing problem can be formulated as a goal-driven MDP over finite state space and a finite set of actions, with each state capturing the position of the sailboat, wind direction, and tack.

In a goal-driven MDP, the lengths of the paths to a terminal state are not necessarily bounded, and thus it is not entirely clear to what depth BRUE should construct its tree. In the Sailing domain, we set H to $4 \times n$, where n is the grid-size of the problem instance, and this because it is unlikely that the optimal path between any two locations on the grid will be longer than a complete encircling of the area.

We compared BRUE with two MCTS-based algorithms: the UCT algorithm, and a recent modification of UCT, obtained from UCT by replacing the UCB1 policy *at the root node* with the uniform policy (Tolpin & Shimony, 2012). In what follows, we denote this modification of UCT as uUCT. The motivation behind the design of uUCT was to improve the empirical simple regret of UCT, and the results for uUCT reported by Tolpin and Shimony (2012) (and confirmed by our experiments here) are impressive. We also display the results for an additional MCTS2e-based algorithm, baptized here as BRucbE, which is very similar to BRUE except that, for exploration, it uses the UCB1 policy instead of the uniform policy. In other words, BRucbE can be seen as "UCT with separation of concerns". All four algorithms were implemented within a single software infrastructure. In line with the setup underlying

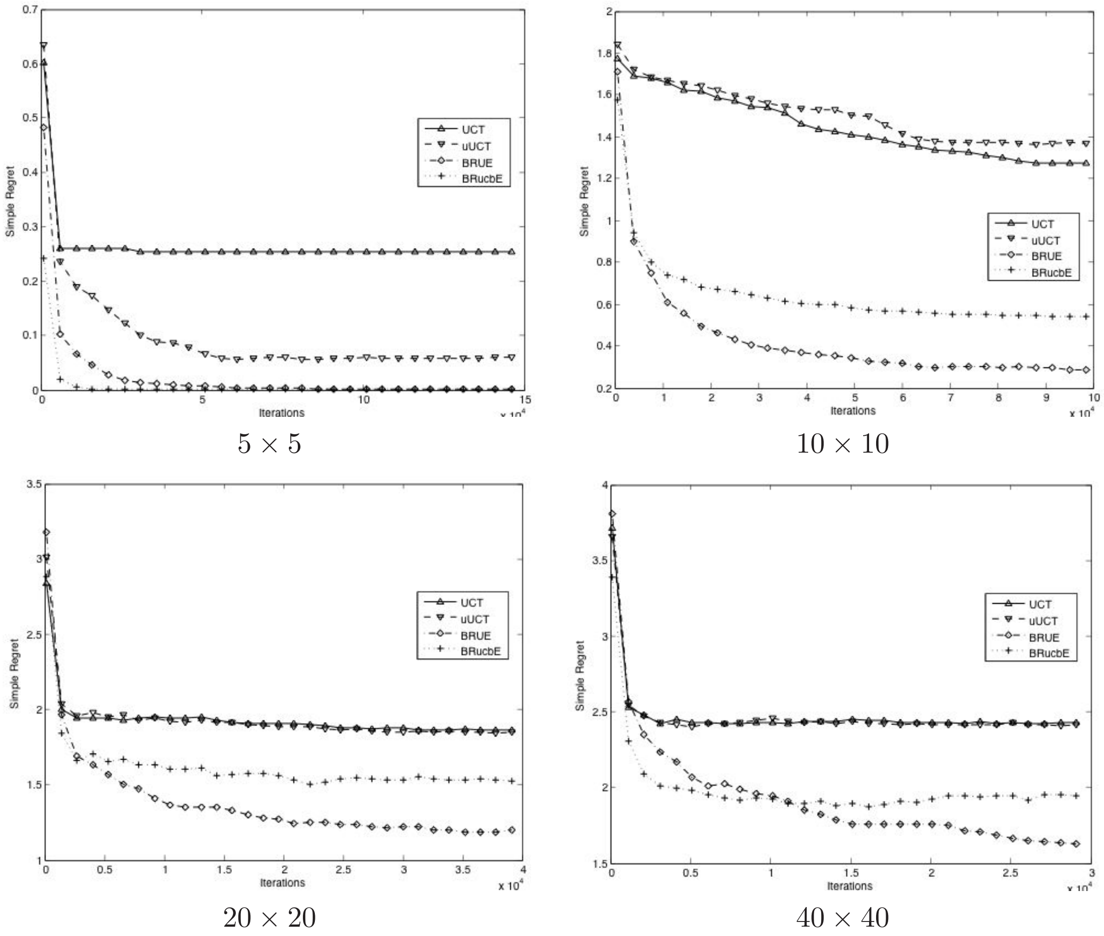
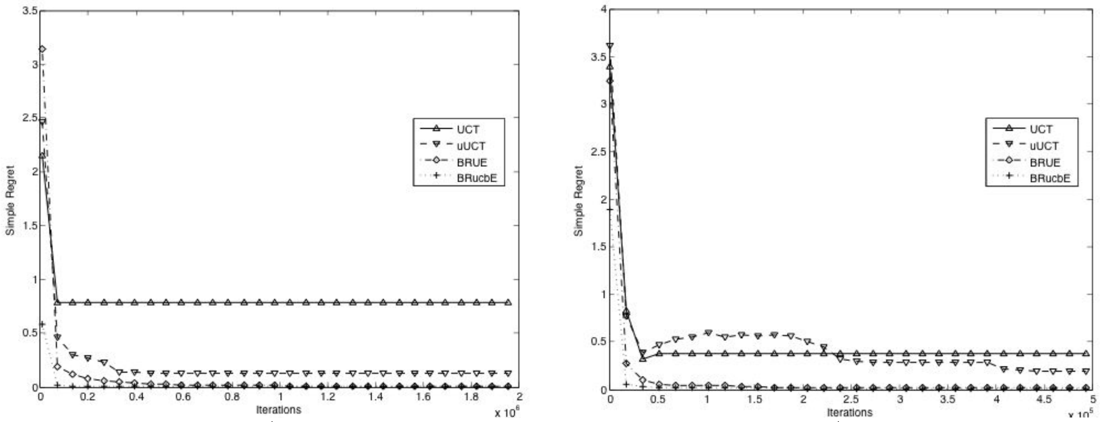


Figure 5: Empirical performance of UCT, uUCT (denoted as UUCT, for short), BRUE, and BRucbE in terms of the average error on the Sailing domain tasks on $n \times n$ grids with $n \in \{5, 10, 20, 40\}$.

Theorem 6 of Kocsis and Szepesvári (2006), the exploration coefficient for UCT and uUCT (parameter c in Eq. 2) was set to the difference between the largest possible and the smallest possible values of the H-step rollouts from the root. In the Sailing domain, this corresponds to the maximal move duration, 6, multiplied by the number of steps-to-go h.

Figure 5 shows the performance of the four algorithms in terms of the empirical simple regret, that is, the average difference $Q(s_0, a) - V(s_0)$ between the true values of the action a chosen by the algorithm and that of the optimal action $\pi^*(s_0)$. Each algorithm was run on 1000 randomly chosen initial states s_0, with the target being fixed at one of the corners of the grid. The performance was measured and depicted as a function of planning time. For all the four algorithms, the planning time unit, or iteration, corresponds to H action samples, that is, to the length of a single rollout.

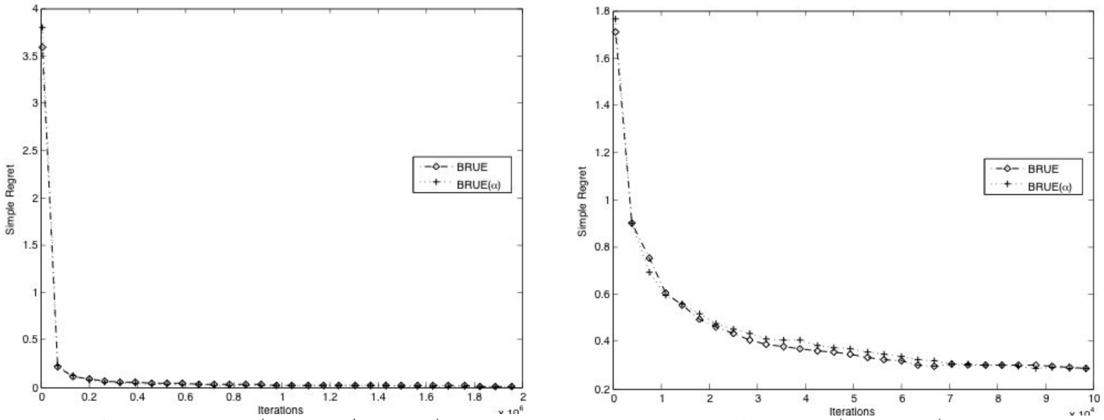


$B = 6/D = 8$ $B = 2/D = 22$

Figure 6: Empirical performance of UCT, uUCT, BRUE, and BRucbE in terms of the average error on the MDP version of random game trees with branching factor B and tree depth D.

Consistently with the results reported in the work of Tolpin and Shimony (2012), on the smaller tasks, uUCT outperformed UCT by a very large margin, with the latter exhibiting very little improvement over time even on the smallest, 5×5, grid. The difference between uUCT and UCT on the larger tasks was less notable. In turn, both BRUE and BRucbE substantially outperformed UCT, with BRucbE being slightly better in smaller tasks, and BRUE taking over in the larger instances, except for relatively short planning deadlines. This shows that the value of MCTS2e's "separation of concerns" lies not only in the ability to employ a pure exploration policy, but also in the ability to base the estimations on the empirically best values, regardless of the employed exploration policy.

Overall, these results on the Sailing domain clearly testify that BRUE is not only attractive in terms of formal guarantees, but can also be very effective in practice. We have also evaluated the four algorithms in a domain of random game trees whose goal is a simple modeling of two-person zero-sum games such as Go, Amazons and Globber. In such games, the winner is decided by a global evaluation of the end board, with the evaluation employing this or another feature counting procedure; the rewards thus are associated only with the terminal states. Following Kocsis and Szepesvári (2006), the rewards in our domain are calculated by first assigning values to moves, and then summing up these values along the paths to the terminal states. Note that the move values are used for the tree construction only and are not made available to the players. The values are chosen uniformly from $[0, 127]$ for the moves of MAX, and from $[-127, 0]$ for the moves of MIN. The players act so to (depending on the role) maximize/minimize their individual payoff: the aim of MAX is to reach terminal s with as high $R(s)$ as possible, and the objective of MIN is similar, *mutatis mutandis*. Our simple game tree model is similar in spirit to many other game tree models used in previous work (Kocsis & Szepesvári, 2006; Smith & Nau, 1994), with two exceptions. First, we measure the success/failure of the players via the actual payoffs they



Game Trees ($B = 6/D = 8$) Sailing (10×10)

Figure 7: Empirical performance of BRUE and BRUE(α) in terms of the average error on the MDP version of random game trees and the sailing domain.

receive, rather than on a ternary scale of win/lose/draw. Moreover, to comply with the setting addressed in this work, we model the game as an MDP where only the moves associated with the MAX player are considered as decision nodes, whereas the moves of MIN are modeled as stochastic outcomes with the following distribution: The optimal minimax move is chosen with probability $p = 0.9$, and the complementary probability $1-p$ is divided uniformly between the rest of the moves.

Similarly to our setup for the Sailing domain, the exploration coefficient for UCT and uUCT was set to the range of the game values, $127H$, since rewards are bounded by the interval $\left[-127\frac{H}{2}, 127\frac{H}{2}\right]$. We ran experiments with two different settings of the branching factor (B) and tree depth (D). As with the Sailing domain, we compared the empirical simple regret obtained by UCT, uUCT, BRUE, and BRucbE over time. Figure 6 shows the performance of the four algorithms for two game configurations, $B = 6, D = 8$ and $B = 2, D = 22$, with each configuration being represented by 1000 game trees. The results here appear encouraging as well, with BRUE and BRucbE overtaking UCT and uUCT, and BRucbE even appearing slightly faster than BRUE in terms of convergence.

We also experimented with BRUE(α) in which, in line with our discussion right after Theorem 3, the α parameter was dynamically adjusted as a function of the depth of the estimated node/action pair. Specifically, we used $\alpha = \frac{n_H}{n_h}$, where n_H denotes the average number of samples of leaf nodes, and n_h denotes the average number of samples of nodes at the same depth of the value estimator under consideration. As we show in Figure 7, we did not find any significant empirical benefit of BRUE(α) over BRUE (to match the superior formal guarantees of the former), neither in the Sailing domain nor in the game trees domain.

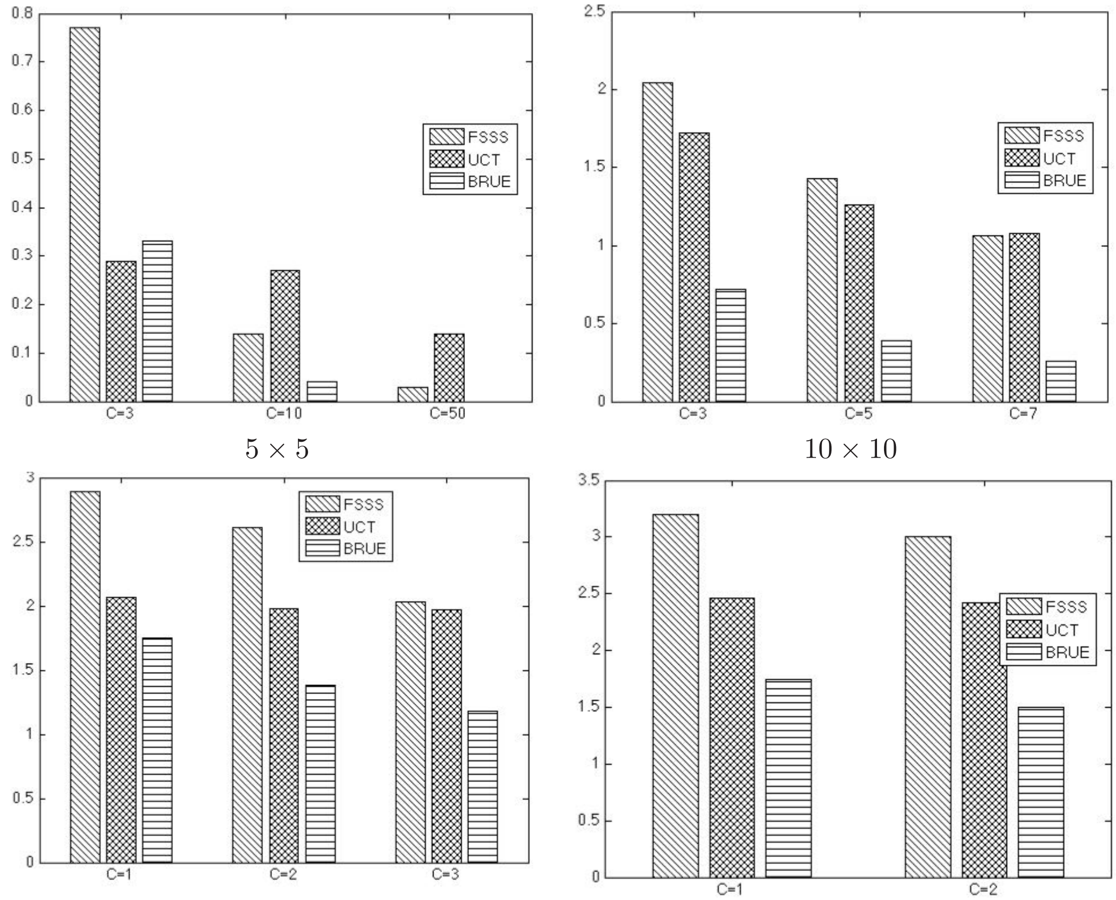
The last set of experiments complements on the theoretical comparison to the sparse sampling (SS) algorithm. Specifically, we performed an empirical comparison between BRUE and UCT with a variant of SS called forward-search sparse sampling (FSSS) (Walsh, Goschin,

& Littman, 2010). Like SS, FSSS estimates the action values at any node using C samples. However, instead of estimating the action values recursively for any encountered state, FSSS uses MCTS-style rollouts to explore the state space, initializing the values of yet unexplored actions with predefined lower and upper bounds. Ultimately, FSSS computes precisely the same values as SS, thus returning the same recommendation. However, it potentially benefits from a kind of pruning to reduce the amount of computation. Notably, unlike SS, FSSS can output an action recommendation at any point of time based on the maintained lower and upper bounds on the actions values. A typical approach is to select the action with the maximum lower bound. However, similarly to SS, FSSS cannot provide any non-trivial guarantees prior to its termination. We therefore choose to use the following experimental setup. First, we run FSSS with some value of C. We then take the overall number of action samples performed by FSSS until termination, and use it as a stopping criteria for BRUE and UCT. Figures 8 and 9 depict the empirical simple regret obtained by the three algorithms upon the termination in the Sailing and game tree domains. For each planning task, we picked a few values of C that allowed FSSS to terminate within a reasonable amount of time.[9] In the Sailing domain, the lower and upper bounds in FSSS were set to 0 and $6h$, respectively, whereas in the game trees domain, we used a lower bound $-127\frac{H}{2}$ and an upper bound $127\frac{H}{2}$.

As it appears, both BRUE and UCT outperform FSSS in most tasks, and notably, BRUE outperforms FSSS in all tasks, and for every value of C. This is despite the purported advantage of FSSS being aware of the termination point. Our explanation for this result concerns two fundamental differences between MCTS-based algorithms and SS. First, recall the formal discussion given after Theorem 1 around the entropy of the transition function. Suppose that FSSS (or SS) estimates a certain action that has two outcomes, with one outcome being more likely than the other. If both outcomes are caught by the C action samples, the same efforts would be invested in estimating the values of these two states, regardless of the fact that one outcome is more likely and thus has a larger contribution to the value of the action. In contrast, both UCT and BRUE adapt to the structure of the problem by skewing the rollouts towards states with higher probability, yielding better results both theoretically and empirically.

Another potential advantage of MCTS algorithms over SS pertains to the allocation of computational efforts to estimating the actions values at different depths. In FSSS (and SS), all the estimations are based on the same number of samples C. In contrast, in both UCT and BRUE, nodes closer to the root are sampled more frequently because of the branching factor. To illustrate the potential benefit of focusing the efforts around the root, let us consider the Sailing domain as an example. The position of the boat reached after taking the optimal moves in the first few steps would probably be closer to the target compared to the position reached after taking non-optimal moves in the fist steps. It is therefore likely that following a random navigation policy from the position of the boat after the first few steps, the target would be reached sooner on average in the former case than in the latter case. In other words, the benefit of knowing the optimal policy at deeper states is smaller, and putting more focus on estimating the actions at nodes closer to the root makes much

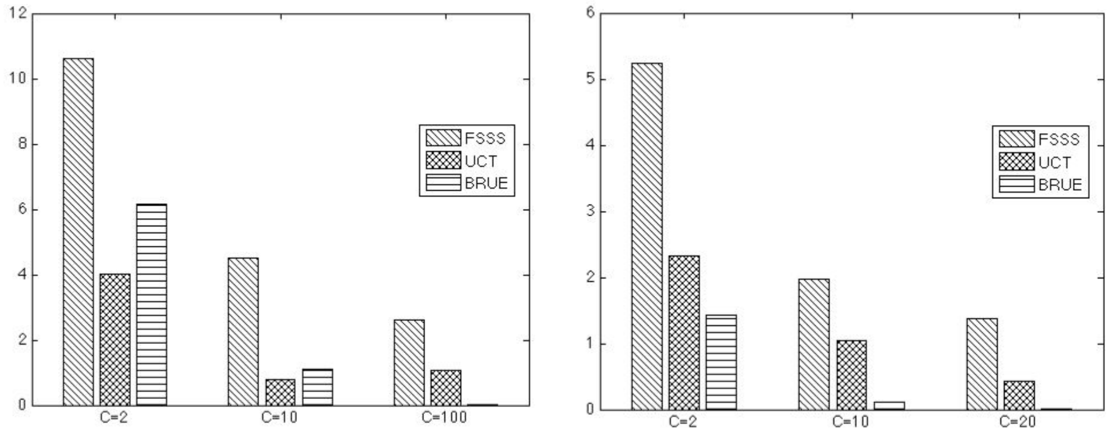
9. The time limit for FSSS was set to 24 hours. Notably, in our implementation of FSSS, we minimize the number of action sample by sampling outcomes only for actions that are selected in the rollouts



5 × 5 | 10 × 10

20 × 20 | 40 × 40

Figure 8: Empirical performance of FSSS, UCT, and BRUE in terms of the average error at termination on the Sailing domain tasks on $n \times n$ grids with $n \in \{5, 10, 20, 40\}$. For $n = 5$, the empirical simple regret of BRUE was 0. For $n = 40$, running FSSS with $C > 1$ took more than 24 hours.



$B = 6/D = 8$ $B = 2/D = 20$

Figure 9: Empirical performance of FSSS, UCT, and BRUE in terms of the average error at termination on the MDP version of random game trees with branching factor B and tree depth D.

sense. We believe that this property prevails in many practical cases, and in these cases MCTS algorithms should be expected to be more efficient.

It is also interesting to see that, although UCT outperforms FSSS in most tasks, the gap between them is decreasing with the size of the budget (C), and in the smaller tasks (Sailing domain with grid 5×5 and 10×10), FSSS even outperforms UCT from some point. We find this to be compliant with the theoretical merits of the pure-exploratory nature of both SS and BRUE.

7. Summary

With the goal of improving the convergence guarantees of smooth Monte-Carlo tree search algorithms for online planning in MDPs, we have introduced a principle of "separation of concerns," as well as a Monte-Carlo tree search scheme, MCTS2e, that allows operationalizing this principle. We showed that a subclass of "purely exploring" instances of MCTS2e guarantees smooth exponential-rate improvement of the performance measures of interest, improving over polynomial-rate guarantees provided by the state-of-the-art algorithms. We then examined, both formally and empirically, a purely exploring MCTS2e algorithm called BRUE. Finally, we explored the prospects of time-dependent "forgetting" of samples within Monte-Carlo search, and showed concrete merits of such sample ignorance on a parametric BRUE(α) algorithm that generalizes BRUE with such "learning with forgetting."

The results open numerous questions for further investigation. First, while BRUE is a rather straightforward implementation of pure exploration with MCTS2e, it is not necessarily the most efficient one. We believe that replacing the uniform exploration of BRUE with a scheme that makes use of the knowledge acquired along the sampling to direct the exploration may result in an empirically more efficient instance of MCTS2e, and possibly even improve on the formal guarantees of BRUE.

Another important point to consider is the speed of convergence to "good" actions, as opposed to the speed of convergence to optimal actions. While BRUE is geared towards identifying the optimal action, "good" is often the best one can hope for when dealing with large MDPs. To identify the optimal solution, BRUE constructs a full-depth tree right from the start. However, focusing on the nodes closer to the root node, e.g., by utilizing more intelligent rules for rollout termination, may improve the quality of the recommendation if the planning time is severely limited. We have recently reported on some successful steps in this direction (Feldman & Domshlak, 2013), but these steps were far from closing this interesting venue of research.

Finally, the core tree sampling scheme employed by BRUE is not the only plausible way to implement the concept of "separation of concerns" discussed in this paper. For instance, substituting the MC-backup procedure with value updates based on Bellman's principle, as, e.g., was done by Keller and Helmert (2013), also constitutes a form of "separation of concerns." It would be interesting to have an in-depth comparison of both the formal and empirical properties of the different protocols.

Acknowledgements

This work is partially supported by and carried out at the Technion-Microsoft Electronic Commerce Research Center, as well as partially supported by the Air Force Office of Scientific Research, USAF, under grant number FA8655-12-1-2096.

Appendix A. Auxiliary Propositions

In the analysis below, we make extensive use of the Hoeffding's tail inequality for sums of bounded independent random variables. In addition, we use the result of the mathematical programming **P1** below.

Hoeffding's tail inequality. Let $X_1, \dots, X_n$ be independent bounded random variables such that X_i falls in the interval $[a_i, b_i]$ with probability 1, and let $S_n = \sum_{i=1}^{n} X_n$. Then, for any $t > 0$, we have

$$\mathbb{P}\left\{S_n - \mathbb{E}S_n \geq t\right\} \leq e^{-2t^2/\sum_{i=1}^{n}(b_i - a_i)^2}.$$

In particular, if all X_i are identically distributed within $[0,1]$ and $\mathbb{E}X_i = \mu$, then

$$\mathbb{P}\left\{S_n - \mu n \geq t\right\} \leq e^{-\frac{2t^2}{n}}.$$

P1 For $h \geq 1$, the solution of the mathematical program

$$\begin{aligned} \underset{p}{\text{maximize}} \quad & \sum_{i=1}^{B} \frac{p_i}{\sqrt[h]{p_i B}} \\ \text{subject to} \quad & \sum_{i=1}^{B} p_i = 1 \\ & 0 \leq p_i \leq 1 \end{aligned}$$

has a value of 1. The result follows from the concavity of the objective function.

Appendix B. Proofs

This appendix is structured in four subsections, respectively dedicated to the proof of Theorem 1, Proposition 4.1.1, and Theorems 2, and 3. For the sake of readability, in places where we believe it does not create any confusion, the expressions of the form $\mathbb{P}(E \mid X_1 = x_1, \ldots, X_k = x_k) \leq f(x_1, \ldots, x_k)$ where X_i are *random variables* are written simply as $\mathbb{P}(E) \leq f(X_1, \ldots, X_k)$.

B.1 Proof of Theorem 1

In what follows, by $V_p^{\pi}(s\langle h\rangle)$ we denote the h-steps value function defined as

$$\mathbb{E}\left[\sum_{i=0}^{h-1} R\left(s_i, \pi\left(s_i\langle h-i\rangle\right), s_{i+1}\right) \,\middle|\, s_0 = s\right],$$

where the expectation is over the transition function Tr, i.e., over the set of conditional probability distributions $\{\mathcal{P}(S \mid s, a)\}_{s,a}$. $V_p^{*}(s\langle h\rangle)$ denotes the value function of the optimal policy π^*. The subscript p is omitted if p corresponds to the transition probabilities $\mathbb{P}$ of the MDP in question.

Theorem 1 follows almost immediately from Lemma 4 below.

Lemma 4 *For any node $s\langle h\rangle$ we have*

$$\mathbb{P}\left\{V^*(s\langle h\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^B}(s\langle h\rangle) \geq \delta\right\} \leq \mathbf{a}_h e^{-\mathbf{b}_h \delta^2 n(s\langle h\rangle)}$$
$$\mathbb{P}\left\{V_{\widehat{\mathcal{P}}}^{\pi^B}(s\langle h\rangle) - V^*(s\langle h\rangle) \geq \delta\right\} \leq \mathbf{a}_h e^{-\mathbf{b}_h \delta^2 n(s\langle h\rangle)},$$

where

$$\mathbf{a}_h = 3K\left(\frac{116 \cdot 9B^2K^2}{\delta^2}\right)^{h-1} (196BK)^{\frac{1}{2}(h-1)^2}(h-1)!^2$$
$$\mathbf{b}_h = \frac{1}{9K^2(196BK)^{h-1}h^2}.$$

Proof: The proof is by induction on h. Starting with $h = 1$, we have

$$
\begin{aligned}
&\mathbb{P}\left\{V^*(s\langle 1\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s\langle 1\rangle) \geq \delta\right\} \\
&\quad \leq \mathbb{P}\left\{Q(s\langle 1\rangle, \pi^*(s\langle 1\rangle)) - Q(s\langle 1\rangle, \pi^{\mathrm{B}}(s\langle 1\rangle)) \geq \frac{2\delta}{3}\right\} \\
&\quad + \mathbb{P}\left\{\sum_{s'}(\mathcal{P}(s' \,|\, s, \pi^{\mathrm{B}}(s\langle 1\rangle)) - \widehat{\mathcal{P}}(s' \,|\, s, \pi^{\mathrm{B}}(s\langle 1\rangle)))R(s, \pi^{\mathrm{B}}(s\langle 1\rangle), s') > \frac{\delta}{3}\right\} \quad \textit{by def. of } Q \\
&\quad \leq \sum_{a \neq \pi^*(s\langle 1\rangle)} \mathbb{P}\left\{\widehat{Q}(s\langle 1\rangle, a) - Q(s\langle 1\rangle, a) \geq \frac{\delta}{3}\right\} \\
&\quad + \mathbb{P}\left\{Q(s\langle 1\rangle, \pi^*(s\langle 1\rangle)) - \widehat{Q}(s\langle 1\rangle, \pi^*(s\langle 1\rangle)) \geq \frac{\delta}{3}\right\} \\
&\quad + \sum_{a \in A(s)} \mathbb{P}\left\{\sum_{s'}(\mathcal{P}(s' \,|\, s, a) - \widehat{\mathcal{P}}(s' \,|\, s, a))R(s, \pi^{\mathrm{B}}(s\langle 1\rangle), s') > \frac{\delta}{3}\right\} \\
&\quad \leq \sum_{a \in A(s)} \mathbb{P}\left\{n(s\langle 1\rangle, a) \leq \frac{n(s\langle 1\rangle)}{2K}\right\} + 2Ke^{-\frac{\delta^2 n(s\langle 1\rangle)}{9K}} \quad \textit{by Hoeffding} \\
&\quad \leq 3Ke^{-\frac{\delta^2 n(s\langle 1\rangle)}{9K^2}}. \quad \textit{by Hoeffding}
\end{aligned}
$$

Assuming now the claim holds for all $h' \leq h$, in proving the induction hypothesis for $h+1$, we encounter the following deficiencies:

(F1) For $h = 1$, $\widehat{Q}$ is an *unbiased* estimator of Q, that is, $\mathbb{E}\widehat{Q} = Q$. In contrast, the estimates inside the tree (at nodes with $h > 1$) are biased. This bias stems from $\widehat{Q}$ possibly being based on numerous sub-optimal choices in the sub-tree rooted in $s\langle h\rangle$.

(F2) For $h = 1$, the summands accumulated by $\widehat{Q}$ are independent. This is not so for $h > 1$, where the accumulated reward depends on the selection of actions in subsequent nodes, which in turn depends on previous rewards.

Our way to circumvent these deficiencies is captured by a sequence of bounding B.1.1-B.1.5 below. At a very high level, we deal with the bias of samples by using a straightforward extension of Hoeffding inequality. In the analysis, the dependence between samples is alleviated by conditioning the outcome of each sample on the state of the information collected at the nodes below the sampled one. All the propositions below are made under the assumption of the induction hypothesis.

Considering a node $s\langle h+1\rangle$, we first show that all the value estimations $\widehat{Q}(s\langle h{+}1\rangle, a)$ of actions $a \in A(s)$ are sufficiently accurate. We show it only for $a = \pi^*(s\langle h+1\rangle)$, whereas the bounds for all other actions can be derived in a similar way. For ease of presentation, in what follows we use the abbreviations $a^* = \pi^*(s\langle h+1\rangle)$, $a^{\mathrm{B}} = \pi^{\mathrm{B}}(s\langle h{+}1\rangle)$, and $n_{a^*} = n(s\langle h+1\rangle, \pi^*(s\langle h+1\rangle))$. We also use the following notation. For $t \in \{1, \ldots, n_{a^*}\}$, let the random variables X_t capture the accumulated reward samples averaged by $\widehat{Q}(s\langle h{+}1\rangle, a^*)$, π_t^{B} capture the policy induced by BRUE at sample t, and $\widehat{\mathcal{P}}_t$ capture the transition probabilities estimations at sample t. In Proposition B.1.2, we bound the error of $\widehat{Q}(s\langle h{+}1\rangle, a^*)$, given that the error of π_t^{B} and $\widehat{\mathcal{P}}_t$ at the descendants of $s\langle h+1\rangle$ during all samples is sufficiently small. In Proposition B.1.1 we bound the probability that the error of π_t^{B} or $\widehat{\mathcal{P}}_t$ during any sample t is too large.

Proposition B.1.1 *For $\delta > 0$, let E_δ be the event in which, while sampling all $X_t, t = 1, \ldots, n_{a^*}$, it holds that,*

1. $\sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) \left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^B}(s'\langle h\rangle)\right) \leq \frac{\delta_t}{2}$, *and*

2. $\sum_{s'} \left(\mathcal{P}(s' \,|\, s, a^*) - \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)\right) (R(s, a^*, s') + V^*(s'\langle h-1\rangle)) \leq \frac{\delta_t}{2}$,

where

$$\delta_t = \sqrt{\frac{\delta^2 n_{a^*}}{9} + \frac{4B \log\left(\frac{n_{a^*}\delta^2 \mathbf{b}_h}{56B}\right)}{\mathbf{b}_h}\frac{1}{\sqrt{t}}}.$$

Then,

$$\mathbb{P}\left\{\neg E_\delta\right\} \leq \frac{112 B^2 \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n_{a^*}}{36B}}. \tag{14}$$

Proof: It follows from **P1** that

$$\begin{aligned} &\mathbb{P}\left\{\sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) \left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle)\right) \geq \frac{\delta_t}{2}\right\} \\ &\quad \leq \sum_{s'} \mathbb{P}\left\{V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle) \geq \frac{\delta_t}{2\sqrt{B\widehat{\mathcal{P}}_t(s' \,|\, s, a^*)}}\right\}. \end{aligned} \tag{15}$$

Indeed, if for all states s' in the summation, it holds that $V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle) < \frac{\delta_t}{2\sqrt{B\widehat{\mathcal{P}}_t(s'|s,a^*)}}$, then, in particular,

$$\begin{aligned} \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) \left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle)\right) &< \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) \frac{\delta_t}{2\sqrt{B\widehat{\mathcal{P}}_t(s' \,|\, s, a^*)}} \\ &= \frac{\delta_t}{2} \sum_{s'} \frac{\widehat{\mathcal{P}}_t(s' \,|\, s, a^*)}{\sqrt{B\widehat{\mathcal{P}}_t(s' \,|\, s, a^*)}} \\ &\leq \frac{\delta_t}{2}. \qquad \textit{by } \textbf{\textit{P1}} \end{aligned}$$

Given that, we have

$$
\begin{aligned}
\mathbb{P}\{\neg E_\delta\} \leq & \sum_{t=1}^{n_{a^*}} \sum_{s'} \mathbb{P}\left\{ V^*(s'\langle h\rangle) - V^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle) > \frac{\delta_t}{2\sqrt{B\widehat{\mathcal{P}}_t(s' \,|\, s, a^*)}} \right\} \\
& + \sum_{t=1}^{n_{a^*}} \mathbb{P}\left\{ \sum_{s'} \left(\mathcal{P}(s' \,|\, s, a^*) - \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)\right) \left(R(s, a^*, s') + V^*(s'\langle h-1\rangle)\right) > \frac{\delta_t}{2} \right\} \\
\leq & \sum_{t=1}^{n_{a^*}} B\mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta_t^2 t}{4B}} \qquad \textit{by I.H.} \\
& + \sum_{t=1}^{n_{a^*}} e^{-\frac{\delta_t^2 t}{4h^2}} \qquad \textit{by Hoeffding} \\
\leq & \sum_{t=1}^{n_{a^*}} 2B\mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta_t^2 t}{4B}} \leq \sum_{t=1}^{n_{a^*}} \frac{112B^2}{n_{a^*}\delta^2 \mathbf{b}_h} \mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta^2 n_{a^*}}{36B}} \qquad \textit{by definition of } \delta_t \\
= & \frac{112B^2 \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n_{a^*}}{36B}}
\end{aligned}
$$

■

Note that, while bounding the probability of the event $\neg E_\delta$ as in Eq. 14, we basically ignore the dependency between the state of π_t^{B} and $\widehat{\mathcal{P}}_t$ during different sampling times, and use a crude union bound. However, if π_t^{B} and $\widehat{\mathcal{P}}_t$ happen to be accurate at some sample t, the probability that they will remain accurate at subsequent samples is higher. It is possible that factoring this dependency into the bound in Proposition B.1.1 can further improve the tightness of the bound.

Conditioned on the state of π_t^{B} and $\widehat{\mathcal{P}}_t$ during all samples $t = 1, \ldots, n_{a^*}$, and given that they are sufficiently accurate as defined by the event E_δ above, Proposition B.1.2 bounds the probability that the value estimator $\widehat{Q}(s\langle h{+}1\rangle, a^*)$ is inaccurate.

Proposition B.1.2 *Under the definition of E_δ introduced in Proposition B.1.1, for all $\delta > 0$, it holds that, given $\{\pi_t^B\}_{t=1}^{n_{a^*}}$, $\{\widehat{\mathcal{P}}_t\}_{t=1}^{n_{a^*}}$, and the event E_δ,*

(1) for all t, the random variables X_t are mutually independent,

(2) for $t \geq 1$, $\mathbb{E}\left[X_t \,\middle|\, \{\pi_t^B\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right] \geq Q(s\langle h{+}1\rangle, a) - \delta_t$, and

(3) $\mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta \,\middle|\, \{\pi_t^B\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \leq e^{-\frac{\delta^2 n_{a^*}}{8(h+1)^2}}$.

Proof: The correctness of mutual independence (1) is direct from the definition of BRUE: all the dependency between the samples in BRUE is induced by the state of the information collected by the samples, and these are determined solely by π^{B} and $\widehat{\mathcal{P}}$. In turn, the proof of (2) is obtained by the definition of E_δ as follows:

$$
\begin{aligned}
\mathbb{E}\left[X_t \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right] &= \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) R(s, a^*, s') + \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h \rangle) \\
&= Q(s\langle h{+}1 \rangle, a^*) \\
&\quad - \sum_{s'} \left(\mathcal{P}(s' \,|\, s, a^*) - \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)\right) \cdot \left(R(s, a^*, s') + V^*(s'\langle h \rangle)\right) \\
&\quad - \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) \left(V^*(s'\langle h \rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h \rangle)\right) \\
&\geq Q(s\langle h{+}1 \rangle, a^*) - \frac{\delta_t}{2} - \frac{\delta_t}{2} \qquad \textit{by definition of } E_\delta \\
&= Q(s\langle h{+}1 \rangle, a^*) - \delta_t
\end{aligned}
$$

Finally, the proof of (3) is obtained by noting that

$$
\begin{aligned}
\frac{1}{n_{a^*}} \sum_{t=1}^{n_{a^*}} \delta_t &= \sqrt{\frac{\delta^2 n_{a^*}}{9} + \frac{4B \log\left(\frac{n_{a^*}\delta^2 \mathbf{b}_h}{56B}\right)}{\mathbf{b}_h}} \cdot \frac{1}{n_{a^*}} \sum_{t=1}^{n_{a^*}} \frac{1}{\sqrt{t}} \\
&\leq \sqrt{\frac{\delta^2 n_{a^*}}{9} + \frac{4B \log\left(\frac{n_{a^*}\delta^2 \mathbf{b}_h}{56B}\right)}{\mathbf{b}_h}} \cdot \frac{2}{\sqrt{n_{a^*}}} \\
&\leq \sqrt{\frac{4\delta^2}{9} + \frac{4\delta^2}{35}} \qquad \textit{since } \frac{\log x}{x} \leq \frac{2}{5} \\
&\leq \frac{3}{4}\delta
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\mathbb{P}\left\{Q(s\langle h{+}1 \rangle, a^*) - \widehat{Q}(s\langle h{+}1 \rangle, a^*) \geq \delta \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \\
&\quad = \mathbb{P}\left\{\mathbb{E}\left[\widehat{Q}(s\langle h{+}1 \rangle, a^*)\right] - \widehat{Q}(s\langle h{+}1 \rangle, a^*) \geq \delta - \frac{1}{n_{a^*}} \sum_{t=1}^{n_{a^*}} \delta_t \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \\
&\quad \leq \mathbb{P}\left\{\mathbb{E}\left[\widehat{Q}(s\langle h{+}1 \rangle, a^*)\right] - \widehat{Q}(s\langle h{+}1 \rangle, a^*) \geq \frac{\delta}{4} \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \\
&\quad \leq e^{-\frac{\cdot \delta^2 n_{a^*}}{8(h+1)^2}},
\end{aligned}
\tag{16}
$$

■

We can now bound the error of the value estimator $\widehat{Q}(s\langle h{+}1 \rangle, a^*)$.

Proposition B.1.3 *For all $\delta > 0$, it holds that*

$$
\mathbb{P}\left\{Q(s\langle h{+}1 \rangle, a^*) - \widehat{Q}(s\langle h{+}1 \rangle, a^*) \geq \delta\right\} \leq \frac{113 B^2 \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n_{a^*}}{36B}}.
$$

Proof:

$$\begin{aligned}
&\mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta\right\} \\
&\quad \leq \mathbb{P}\left\{\neg E_\delta\right\} \\
&\qquad + \sum_{\{\pi_t^{\mathrm{B}}, \widehat{\mathcal{P}}_t\}} \mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \mathbb{P}\left\{\{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\} \;\middle|\; E_\delta\right\} \\
&\quad \leq \frac{113 B^2 \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n_{a^*}}{36B}} \qquad \textit{by Props. B.1.1 \& B.1.2.}
\end{aligned}$$

■

Proposition B.1.4 below employs the bounds on the accuracy of $\widehat{Q}(s\langle h{+}1\rangle, a)$ to bound the simple regret of a^{B}.

Proposition B.1.4

$$\mathbb{P}\left\{Q(s\langle h{+}1\rangle, \pi^*(s\langle h{+}1\rangle)) - Q(s\langle h{+}1\rangle, a^B) \geq \delta\right\} \quad \leq \quad \frac{114 B^2 \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n(s\langle h+1\rangle)}{144BK}}$$

Proof:

$$\begin{aligned}
&\mathbb{P}\left\{Q(s\langle h{+}1\rangle, \pi^*(s\langle h{+}1\rangle)) - Q(s\langle h{+}1\rangle, a^{\mathrm{B}}) \geq \delta\right\} \\
&\quad \leq \sum_{a \neq \pi^*(s\langle h+1\rangle)} \mathbb{P}\left\{\widehat{Q}(s\langle h{+}1\rangle, a) - Q(s\langle h{+}1\rangle, a) \geq \frac{\delta}{2}\right\} \\
&\quad + \mathbb{P}\left\{Q(s\langle h{+}1\rangle, \pi^*(s\langle h{+}1\rangle)) - \widehat{Q}(s\langle h{+}1\rangle, \pi^*(s\langle h{+}1\rangle)) \geq \frac{\delta}{2}\right\} \\
&\quad \leq \sum_{a \in A(s)} \mathbb{P}\left\{n(s\langle h+1\rangle, a) \leq \frac{n(s\langle h+1\rangle)}{2K}\right\} + \frac{113 B^2 K \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n(s\langle h+1\rangle)}{144BK}} \qquad \textit{by Prop. B.1.3} \\
&\quad \leq \frac{114 B^2 K \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n(s\langle h+1\rangle)}{144BK}} \qquad \textit{by Hoeffding}
\end{aligned}$$

■

The induction step is concluded by Proposition B.1.5.

Proposition B.1.5

$$\mathbb{P}\left\{V^*(s\langle h{+}1\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^B}(s\langle h{+}1\rangle) \geq \delta\right\} \leq \frac{116 B^2 K \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n(s\langle h+1\rangle)}{196BK}}.$$

Proof: Since we have

$$\begin{aligned}
&V^*(s\langle h{+}1\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s\langle h{+}1\rangle) \\
&\quad = Q(s\langle h{+}1\rangle, \pi^*(s\langle h{+}1\rangle)) - Q(s\langle h{+}1\rangle, a^{\mathrm{B}}) \\
&\qquad + \sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s, a^{\mathrm{B}}) \left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s'\langle h\rangle)\right) \\
&\qquad + \sum_{s'} \left(\mathcal{P}(s' \,|\, s, a^{\mathrm{B}}) - \widehat{\mathcal{P}}(s' \,|\, s, a^{\mathrm{B}})\right) \left(R(s, a^{\mathrm{B}}, s') + V^*(s'\langle h\rangle)\right),
\end{aligned}$$

it holds that

$$
\begin{aligned}
&\mathbb{P}\left\{V^*(s\langle h{+}1\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s\langle h{+}1\rangle) \geq \delta\right\} \\
&\quad\leq \mathbb{P}\left\{Q(s\langle h{+}1\rangle, \pi^*(s\langle h{+}1\rangle)) - Q(s\langle h{+}1\rangle, a^{\mathrm{B}}) \geq \frac{6\delta}{7}\right\} \\
&\qquad+ \mathbb{P}\left\{\sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s, a^{\mathrm{B}}) \left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s'\langle h\rangle)\right) \geq \frac{\delta}{14}\right\} \\
&\qquad+ \mathbb{P}\left\{\sum_{s'} \left(\mathcal{P}(s' \,|\, s, a^{\mathrm{B}}) - \widehat{\mathcal{P}}(s' \,|\, s, a^{\mathrm{B}})\right) \left(R(s, a^{\mathrm{B}}, s') + V^*(s'\langle h\rangle)\right) > \frac{\delta}{14}\right\} \\
&\quad\leq \frac{114B^2K\mathbf{a}_h}{\delta^2\mathbf{b}_h} e^{-\frac{\mathbf{b}_h\delta^2 n(s\langle h+1\rangle)}{144BK}} \qquad \textit{by Prop. B.1.4} \\
&\qquad+ \sum_{a\in A(s)} \mathbb{P}\left\{\sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s, a) \left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s'\langle h\rangle)\right) \geq \frac{\delta}{14}\right\} \\
&\qquad+ \sum_{a\in A(s)} \mathbb{P}\left\{\sum_{s'} \left(\mathcal{P}(s' \,|\, s, a) - \widehat{\mathcal{P}}(s' \,|\, s, a)\right) \left(R(s, a, s') + V^*(s'\langle h\rangle)\right) > \frac{\delta}{14}\right\} \\
&\quad\leq \frac{114B^2K\mathbf{a}_h}{\delta^2\mathbf{b}_h} e^{-\frac{\mathbf{b}_h\delta^2 n(s\langle h+1\rangle)}{196BK}} + BK\mathbf{a}_h e^{-\frac{\mathbf{b}_h\delta^2 n(s\langle h+1\rangle)}{196BK}} + Ke^{-\frac{-\delta^2 n(s\langle h+1\rangle)}{196K(h+1)^2}} \\
&\quad\leq \frac{116B^2K\mathbf{a}_h}{\delta^2\mathbf{b}_h} e^{-\frac{\mathbf{b}_h h^2\delta^2 n(s\langle h+1\rangle)}{196BK(h+1)^2}}.
\end{aligned}
$$

Proving the bound for $\mathbb{P}\left\{V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s\langle h{+}1\rangle) - V^*(s\langle h{+}1\rangle) \geq \delta\right\}$ is completely similar. ■

Finally, the proof of Theorem 1 is concluded by

$$
\begin{aligned}
&\mathbb{P}\left\{\pi_n^{\mathrm{B}}(s_0\langle H\rangle) \neq \pi^*(s_0\langle H\rangle)\right\} \\
&\quad\leq \mathbb{P}\left\{Q(s_0\langle H\rangle, \pi^*(s_0\langle H\rangle)) - Q(s_0\langle H\rangle, \pi^{\mathrm{B}}(s_0\langle H\rangle)) \geq \Delta\right\} \\
&\quad\leq 3K\left(\frac{116\cdot 9B^2K^2}{\Delta^2}\right)^{H-1} (196BK)^{\frac{1}{2}(H-1)^2}(H-1)!^2 e^{-\frac{\Delta^2 n}{9K^2(196BK)^{H-1}H^2}}
\end{aligned}
\tag{17}
$$

and by noting that the maximal loss from choosing a sub-optimal action at $s_0\langle H\rangle$ is H. ■

B.2 Proof of Proposition 4.1.1 (BRUE bounds with P_e)

Basically, the proof for Proposition 4.1.1 is identical to that of Theorem 1 for $h < 4$. For $h \geq 4$, we note that

$$
\begin{aligned}
\mathbb{P}\left\{\sum_{s'} \sqrt{\widehat{\mathcal{P}}_t(s' \,|\, s, a)} > 3\sqrt{P_e}\right\} &\leq \sum_{s'} \mathbb{P}\left\{\widehat{\mathcal{P}}_t(s' \,|\, s, a) > \mathcal{P}(s' \,|\, s, a) + \frac{P_e}{B^2}\right\} \\
&\leq Be^{-\frac{2t}{B^4}}. \qquad \textit{by Hoeffding}
\end{aligned}
\tag{18}
$$

Indeed, if $\widehat{\mathcal{P}}_t(s' \,|\, s, a) \leq \mathcal{P}(s' \,|\, s, a) + \frac{P_e}{B^2}$ for all s', then

$$\begin{aligned}\sum_{s'} \sqrt{\widehat{\mathcal{P}}_t(s' \,|\, s, a)} &\leq \sum_{s'} \sqrt{\mathcal{P}(s' \,|\, s, a) + \frac{P_e}{B^2}} \\ &\leq \sum_{s'} \left[\sqrt{2\mathcal{P}(s' \,|\, s, a)} + \sqrt{2\frac{P_e}{B^2}} \right] \\ &\leq 2\sqrt{2P_e} \leq 3\sqrt{P_e}\end{aligned}$$

Therefore, the probability in Eq. 15 from Proposition B.1.1 can be bounded for $h > 4$ as

$$\begin{aligned}&\mathbb{P}\left\{ \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) \left(V^*(s'\langle h \rangle) - V^{\pi_t^{\mathrm{B}}}_{\widehat{\mathcal{P}}_t}(s'\langle h \rangle) \right) \geq \frac{\delta_t}{2} \right\} \\ &\leq \sum_{s'} \mathbb{P}\left\{ V^*(s'\langle h \rangle) - V^{\pi_t^{\mathrm{B}}}_{\widehat{\mathcal{P}}_t}(s'\langle h \rangle) \geq \frac{\delta_t}{6\sqrt{P_e \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)}} \right\} + \mathbb{P}\left\{ \sum_{s'} \sqrt{\widehat{\mathcal{P}}_t(s' \,|\, s, a)} > 3\sqrt{P_e} \right\} \\ &\leq B\mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta_t^2 t}{36 P_e}} + B e^{-\frac{2t}{B^4}} \qquad \textit{by I.H. \& Eq. 18} \\ &\leq 2B\mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta_t^2 t}{36 P_e}}.\end{aligned}$$

Consequently, we have that

$$\mathbb{P}\left\{\neg E_\delta\right\} \leq \frac{158 B^2 \mathbf{a}_h}{\delta^2 \mathbf{b}_h} e^{-\frac{\mathbf{b}_h \delta^2 n_{a^*}}{306 P_e}}.$$

Plugging this bound for $\mathbb{P}\left\{\neg E_\delta\right\}$ in the chain of bounding in Propositions B.1.2-B.1.5, we obtain the result.

B.3 Proof for Theorem 2

The proof of Theorem 2 follows from the proof of Theorem 1 and noting that

- the effect of the ROLLOUT-ACTION policy on the convergence rate comes into play only in the bounds on $\mathbb{P}\left\{ n(s\langle h{+}1 \rangle, a) < \frac{n(s\langle h+1 \rangle)}{2K} \right\}$, and
- the condition of Theorem 2 simply postulates such bounds so that the exponential-rate convergence is guaranteed.

B.4 Proof for Theorem 3

In general, the proof of Theorem 3 follows the same lines as the proof of convergence rate for BRUE in Theorem 1, with the role of each Proposition B.3.i here corresponding to the role of Proposition B.1.i in the proof of Theorem 1. Essentially, the proof of Theorem 3 deviates substantially from the proof of Theorem 1 only in the modification of Propositions B.1.1 and B.1.2 to partial averaging, captured by Propositions B.4.1 and B.4.2, respectively. The bounds in the rest of the propositions are adjusted accordingly. We formulate the proof for arbitrary values of α, although we derive the bounds for a particular choice of depth-dependent $\alpha_h \approx \frac{1}{(BK)^{h-1}}$. Similarly to Theorem 1, the proof for Theorem 3 is based on Lemma 5 below.

Lemma 5 *For any node $s\langle h\rangle$ we have*

$$\mathbb{P}\left\{V^*(s\langle h\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^B}(s\langle h\rangle) \geq \delta\right\} \leq \mathbf{a}_h e^{-\mathbf{b}_h \delta^2 n(s\langle h\rangle)}$$
$$\mathbb{P}\left\{V_{\widehat{\mathcal{P}}}^{\pi^B}(s\langle h\rangle) - V^*(s\langle h\rangle) \geq \delta\right\} \leq \mathbf{a}_h e^{-\mathbf{b}_h \delta^2 n(s\langle h\rangle)},$$

where

$$\mathbf{a}_h = 3K\left(\frac{12BK}{\delta^2}\right)^{h-1}(h!)^2,$$
$$\mathbf{b}_h = \frac{1}{9K^2(196BK)^{h-1}h^2}.$$

Proof: The proof is by induction on h. The base of the induction is identical to that in Lemma 4, so we continue straight with the induction step. All the propositions below are made under the assumption of the induction hypothesis. Considering a node $s\langle h+1\rangle$, we make use of the same notation used in the proof for Theorem 1, namely, $a^* = \pi^*(s\langle h+1\rangle)$, $a^{\mathrm{B}} = \pi^{\mathrm{B}}(s\langle h{+}1\rangle)$, $n_{a^*} = n(s\langle h+1\rangle, \pi^*(s\langle h+1\rangle))$, and, for $t \in \{1, \ldots, n_{a^*}\}$, the random variables X_t capture the accumulated reward samples averaged by $\widehat{Q}(s\langle h{+}1\rangle, a^*)$, π_t^{B} capture the policy induced by BRUE at sample t, and $\widehat{\mathcal{P}}_t$ capture the transition probabilities estimations at sample t. In addition, we also use the additional abbreviation $n_{a^*}^{\alpha} = \lfloor(1{-}\alpha)n_{a^*}\rfloor$.

Proposition B.4.1 *For $\delta > 0$, let E_δ be the event in which, while sampling $X_t, t = n_{a^*}^{\alpha}, \ldots, n_{a^*}$, it holds that*

1. $\sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)\left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle)\right) \leq \frac{\delta_t}{2}$, *and*
2. $\sum_{s'}\left(\mathcal{P}(s' \,|\, s, a^*) - \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)\right)(R(s, a^*, s') + V^*(s'\langle h-1\rangle)) \leq \frac{\delta_t}{2}$,

where

$$\delta_t = \sqrt{\frac{\delta^2 n_{a^*}}{3} + \frac{4B\log\left(\frac{\alpha_{h+1}n_{a^*}\delta^2}{8h^2}\right)}{\mathbf{b}_h}}\frac{1}{\sqrt{t}}.$$

Then,

$$\mathbb{P}\left\{\neg E_\delta\right\} \leq \frac{8Bh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h\delta^2 n_{a^*}}{12B}}.$$

Proof: It follows from **P1** that

$$\mathbb{P}\left\{\sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)\left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle)\right) \geq \frac{\delta_t}{2}\right\}$$
$$\leq \sum_{s'}\mathbb{P}\left\{V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle) \geq \frac{\delta_t}{2\sqrt{B\widehat{\mathcal{P}}_t(s' \,|\, s, a^*)}}\right\}$$

Thus,

$$\begin{aligned}
\mathbb{P}\left\{\neg E_\delta\right\} \leq & \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} \sum_{s'} \mathbb{P}\left\{V^*(s'\langle h\rangle) - V^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle) > \frac{\delta_t}{2\sqrt{B\widehat{\mathcal{P}}_t(s' \,|\, s, a^*)}}\right\} \\
& + \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} \mathbb{P}\left\{\sum_{s'}\left(\mathcal{P}(s' \,|\, s, a^*) - \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)\right)\left(R(s, a^*, s') + V^*(s'\langle h-1\rangle)\right) > \frac{\delta_t}{2}\right\} \\
\leq & \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} B\mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta_t^2 t}{4B}} \qquad \textit{by I.H.} \\
& + \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} e^{-\frac{\delta_t^2 t}{4h^2}} \qquad \textit{by Hoeffding} \\
\leq & \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} 2B\mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta_t^2 t}{4B}} \leq \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} \frac{8Bh^2}{\alpha_{h+1} n_{a^*} \delta^2} \mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta^2 n_{a^*}}{12B}} \qquad \textit{by definition of } \delta_t \\
= & \frac{8Bh^2}{\delta^2} \mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta^2 n_{a^*}}{12B}}
\end{aligned}$$

∎

Prop. B.4.2 is modified accordingly.

Proposition B.4.2 *Under the definition of E_δ introduced in Proposition B.4.1, for all $\delta > 0$, it holds that, given $\{\pi_t^B\}_{t=1}^{n_{a^*}}$, $\{\widehat{\mathcal{P}}_t\}_{t=1}^{n_{a^*}}$, and the event E_δ,*

(1) for all t, the random variables X_t are mutually independent,

(2) for $t \geq n_{a^*}^{\alpha}$, $\mathbb{E}\left[X_t \,\middle|\, \{\pi_t^B\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right] \geq Q(s\langle h{+}1\rangle, a) - \delta_t$, and

(3) $\mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta \,\middle|\, \{\pi_t^B\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \leq e^{-\frac{\mathbf{b}_h h^2 \delta^2 n_{a^*}}{8B(h+1)^2}}$.

Proof: The correctness of mutual independence (1) is direct from the definition of BRUE: all the dependency between the samples in BRUE is induced by the state of the information collected by the samples, and these are determined solely by π^{B} and $\widehat{\mathcal{P}}$. In turn, the proof of (2) is obtained by the definition of E_δ as follows:

$$
\begin{aligned}
\mathbb{E}\left[X_t \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right] &= \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) R(s, a^*, s') + \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle) \\
&= Q(s\langle h{+}1\rangle, a^*) \\
&\quad - \sum_{s'} \left(\mathcal{P}(s' \,|\, s, a^*) - \widehat{\mathcal{P}}_t(s' \,|\, s, a^*)\right) \cdot \left(R(s, a^*, s') + V^*(s'\langle h\rangle)\right) \\
&\quad - \sum_{s'} \widehat{\mathcal{P}}_t(s' \,|\, s, a^*) \left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}_t}^{\pi_t^{\mathrm{B}}}(s'\langle h\rangle)\right) \\
&\geq Q(s\langle h{+}1\rangle, a^*) - \frac{\delta_t}{2} - \frac{\delta_t}{2} \qquad \textit{by definition of } E_\delta \\
&= Q(s\langle h{+}1\rangle, a^*) - \delta_t
\end{aligned}
$$

Finally, the proof of (3) is obtained by noting that

$$
\begin{aligned}
\frac{1}{\alpha_{h+1} n_{a^*}} \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} \delta_t &= \sqrt{\frac{\delta^2 n_{a^*}}{3} + \frac{4B \log\left(\frac{\alpha_{h+1} n_{a^*} \delta^2}{8h^2}\right)}{\mathbf{b}_h}} \cdot \frac{1}{\alpha_{h+1} n_{a^*}} \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} \frac{1}{\sqrt{t}} \\
&\leq \sqrt{\frac{\delta^2 n_{a^*}}{3} + \frac{4B \log\left(\frac{\alpha_{h+1} n_{a^*} \delta^2}{8h^2}\right)}{\mathbf{b}_h}} \cdot \frac{2}{\sqrt{n_{a^*}}} \cdot \frac{1 - \sqrt{1 - \alpha_{h+1}}}{\alpha_{h+1}} \\
&\leq \sqrt{\frac{4\delta^2}{9} + \frac{4\delta^2}{35}} \qquad \textit{since } \frac{\log x}{x} \leq \frac{2}{5} \textit{ and } \alpha_{h+1} = \frac{\mathbf{b}_h h^2}{B} \\
&\leq \frac{3}{4}\delta
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \\
&\quad = \mathbb{P}\left\{\mathbb{E}\left[\widehat{Q}(s\langle h{+}1\rangle, a^*)\right] - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta - \frac{1}{\alpha_{h+1} n_{a^*}} \sum_{t=n_{a^*}^{\alpha}}^{n_{a^*}} \delta_t \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \\
&\quad \leq \mathbb{P}\left\{\mathbb{E}\left[\widehat{Q}(s\langle h{+}1\rangle, a^*)\right] - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \frac{\delta}{4} \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \\
&\quad \leq e^{-\frac{\mathbf{b}_h h^2 \delta^2 n_{a^*}}{8B(h+1)^2}}.
\end{aligned} \tag{19}
$$

■

Proposition B.4.3 *For all $\delta > 0$, it holds that*

$$
\mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta\right\} \leq \frac{9Bh^2}{\delta^2} \mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2 \delta^2 n_{a^*}}{12B(h+1)^2}}.
$$

Proof:

$$
\begin{aligned}
&\mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta\right\} \\
&\quad \leq \mathbb{P}\left\{\neg E_\delta\right\} \\
&\qquad + \sum_{\{\pi_t^{\mathrm{B}}, \widehat{\mathcal{P}}_t\}} \mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \delta \;\middle|\; \{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\}, E_\delta\right\} \mathbb{P}\left\{\{\pi_t^{\mathrm{B}}\}, \{\widehat{\mathcal{P}}_t\} \;\middle|\; E_\delta\right\} \\
&\quad \leq \frac{9Bh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2 \delta^2 n_{a^*}}{12B(h+1)^2}} \qquad \textit{by Props. B.4.1 \& B.4.2.}
\end{aligned}
$$

■

The remainder of the proof is identical to the proof of Theorem 1, whereas only the bounds on the error of the estimators $\widehat{Q}(s\langle h{+}1\rangle, a)$ are aligned with Proposition B.4.3.

Proposition B.4.4

$$
\mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - Q(s\langle h{+}1\rangle, a^B) \geq \delta\right\} \quad \leq \quad \frac{10BKh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2 \delta^2 n(s\langle h+1\rangle)}{96BK(h+1)^2}}
$$

Proof:

$$
\begin{aligned}
&\mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - Q(s\langle h{+}1\rangle, a^{\mathrm{B}}) \geq \delta\right\} \\
&\quad \leq \sum_{a \neq a^*} \mathbb{P}\left\{\widehat{Q}(s\langle h{+}1\rangle, a) - Q(s\langle h{+}1\rangle, a) \geq \frac{\delta}{2}\right\} + \mathbb{P}\left\{Q(s\langle h{+}1\rangle, a^*) - \widehat{Q}(s\langle h{+}1\rangle, a^*) \geq \frac{\delta}{2}\right\} \\
&\quad \leq \sum_{a \in A(s)} \mathbb{P}\left\{n(s\langle h+1\rangle, a) \leq \frac{n(s\langle h+1\rangle)}{2K}\right\} + \frac{9BKh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2 \delta^2 n(s\langle h+1\rangle)}{96BK(h+1)^2}} \qquad \textit{by Prop. B.4.3} \\
&\quad \leq \frac{10BKh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2 \delta^2 n(s\langle h+1\rangle)}{96BK(h+1)^2}}. \qquad \textit{by Hoeffding}
\end{aligned}
$$

■

The induction step is concluded by Proposition B.4.5.

Proposition B.4.5

$$
\mathbb{P}\left\{V^*(s\langle h{+}1\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^B}(s\langle h{+}1\rangle) \geq \delta\right\} \leq \frac{12BKh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2 \delta^2 n(s\langle h+1\rangle)}{196BK(h+1)^2}}.
$$

Proof: Since we have

$$
\begin{aligned}
&V^*(s\langle h{+}1\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s\langle h{+}1\rangle) \\
&\quad = Q(s\langle h{+}1\rangle, \pi^*(s\langle h{+}1\rangle)) - Q(s\langle h{+}1\rangle, a^{\mathrm{B}}) \\
&\qquad + \sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s, a^{\mathrm{B}}) \left(V^*(s'\langle h\rangle) - V_{\widehat{\mathcal{P}}}^{\pi^{\mathrm{B}}}(s'\langle h\rangle)\right) \\
&\qquad + \sum_{s'} \left(\mathcal{P}(s' \,|\, s, a^{\mathrm{B}}) - \widehat{\mathcal{P}}(s' \,|\, s, a^{\mathrm{B}})\right) \left(R(s, a^{\mathrm{B}}, s') + V^*(s'\langle h\rangle)\right),
\end{aligned}
$$

it holds that

$$\begin{aligned}
&\mathbb{P}\left\{V^*(s\langle h{+}1\rangle) - V^{\pi^{\mathrm{B}}}_{\widehat{\mathcal{P}}}(s\langle h{+}1\rangle) \geq \delta\right\} \\
&\quad \leq \mathbb{P}\left\{Q(s\langle h{+}1\rangle, \pi^*(s\langle h{+}1\rangle)) - Q(s\langle h{+}1\rangle, a^{\mathrm{B}}) \geq \frac{6\delta}{7}\right\} \\
&\qquad + \mathbb{P}\left\{\sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s, a^{\mathrm{B}}) \left(V^*(s'\langle h\rangle) - V^{\pi^{\mathrm{B}}}_{\widehat{\mathcal{P}}}(s'\langle h\rangle)\right) \geq \frac{\delta}{14}\right\} \\
&\qquad + \mathbb{P}\left\{\sum_{s'} \left(\mathcal{P}(s' \,|\, s, a^{\mathrm{B}}) - \widehat{\mathcal{P}}(s' \,|\, s, a^{\mathrm{B}})\right) \left(R(s, a^{\mathrm{B}}, s') + V^*(s'\langle h\rangle)\right) > \frac{\delta}{14}\right\} \\
&\quad \leq \frac{10BKh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2\delta^2 n(s\langle h+1\rangle)}{136BK(h+1)^2}} \qquad \textit{by Prop. B.4.4} \\
&\qquad + \sum_{a\in A(s)} \mathbb{P}\left\{\sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s, a) \left(V^*(s'\langle h\rangle) - V^{\pi^{\mathrm{B}}}_{\widehat{\mathcal{P}}}(s'\langle h\rangle)\right) \geq \frac{\delta}{14}\right\} \\
&\qquad + \sum_{a\in A(s)} \mathbb{P}\left\{\sum_{s'} \left(\mathcal{P}(s' \,|\, s, a) - \widehat{\mathcal{P}}(s' \,|\, s, a)\right) \left(R(s, a, s') + V^*(s'\langle h\rangle)\right) > \frac{\delta}{14}\right\} \\
&\quad \leq \frac{10BKh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2\delta^2 n(s\langle h+1\rangle)}{136BK(h+1)^2}} + BK\mathbf{a}_h e^{-\frac{\mathbf{b}_h \delta^2 n(s\langle h+1\rangle)}{196BK}} + Ke^{-\frac{-\delta^2 n(s\langle h+1\rangle)}{196K(h+1)^2}} \\
&\quad \leq \frac{12BKh^2}{\delta^2}\mathbf{a}_h e^{-\frac{\mathbf{b}_h h^2\delta^2 n(s\langle h+1\rangle)}{196BK(h+1)^2}}.
\end{aligned}$$

Proving the bound for $\mathbb{P}\left\{V^{\pi^{\mathrm{B}}}_{\widehat{\mathcal{P}}}(s\langle h{+}1\rangle) - V^*(s\langle h{+}1\rangle) \geq \delta\right\}$ is completely similar. ■

Finally, the proof of Theorem 3 is concluded by

$$\begin{aligned}
&\mathbb{P}\left\{\pi^{\mathrm{B}}_n(s_0\langle H\rangle) \neq \pi^*(s_0\langle H\rangle)\right\} \\
&\quad \leq \mathbb{P}\left\{Q(s_0\langle H\rangle, \pi^*(s_0\langle H\rangle)) - Q(s_0\langle H\rangle, \pi^{\mathrm{B}}(s_0\langle H\rangle)) \geq \Delta\right\} \\
&\quad \leq 3K\left(\frac{12BK}{\Delta^2}\right)^{H-1}(H!)^2 e^{-\frac{\Delta^2 n}{9K^2(196BK)^{H-1}H^2}}
\end{aligned} \tag{20}$$

and by noting that the maximal loss from choosing a sub-optimal action at $s_0\langle H\rangle$ is H. ■

Appendix C. Proof for Proposition 2.1.1 (SS bound)

Let $\widehat{Q}(s\langle h\rangle, a)$ be the average of C recursive samples of the value of action a in $s\langle h\rangle$, and let $\widehat{\mathcal{P}}(s' \mid s, a)$ be the empirical transition probability based on these C samples. For any node $s\langle h\rangle$ and action a, with probability at least $1 - e^{-\frac{2\delta^2 C}{H^2}}$, we have

$$\left|\sum_{s'} \left(\mathcal{P}(s' \,|\, s, a) - \widehat{\mathcal{P}}(s' \,|\, s, a)\right) V^*(s'\langle h-1\rangle)\right| \leq \delta$$

Since sparse sampling encounters at most $(\min(B,C)\cdot K)^H$ nodes, we have that the probability of some bad estimate is bounded by $(\min(B,C)\cdot K)^H e^{-\frac{2\delta^2 C}{H^2}}$. Therefore, we have

$$\begin{aligned}
&Q(s\langle h\rangle, a) - \widehat{Q}(s\langle h\rangle, a) \\
&\qquad \leq \sum_{s'} \left(\mathcal{P}(s' \,|\, s,a) - \widehat{\mathcal{P}}(s' \,|\, s,a)\right) V^*(s'\langle h-1\rangle) \\
&\qquad + \sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s,a) \left(Q(s'\langle h-1\rangle, \pi^*(s'\langle h-1\rangle)) - \max_{a'} \widehat{Q}(s'\langle h-1\rangle, a')\right) \\
&\qquad \leq \delta + \sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s,a) \left(Q(s'\langle h-1\rangle, \pi^*(s'\langle h-1\rangle)) - \widehat{Q}(s'\langle h-1\rangle, \pi^*(s'\langle h-1\rangle))\right),
\end{aligned} \tag{21}$$

and similarly,

$$\begin{aligned}
&\widehat{Q}(s\langle h\rangle, a) - Q(s\langle h\rangle, a) \\
&\qquad \leq \sum_{s'} \left(\widehat{\mathcal{P}}(s' \,|\, s,a) - \mathcal{P}(s' \,|\, s,a)\right) V^*(s'\langle h-1\rangle) \\
&\qquad + \sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s,a) \left(\max_{a'} \widehat{Q}(s'\langle h-1\rangle, a') - Q(s'\langle h-1\rangle, \pi^*(s'\langle h-1\rangle))\right) \\
&\qquad \leq \delta + \sum_{s'} \widehat{\mathcal{P}}(s' \,|\, s,a) \max_{a'} \left\{\widehat{Q}(s'\langle h-1\rangle, a') - Q(s'\langle h-1\rangle, a')\right\}.
\end{aligned} \tag{22}$$

Note that the two bounds in Equations 21,22 above result in the recursion $\alpha_h = \delta + \alpha_{h-1} = \delta h$, where α_h upper bounds the respective difference. This implies that, with probability at most $(\min(B,C)\cdot K)^H e^{-\frac{2\delta^2 C}{H^2}}$,

$$\left|Q(s_0\langle H\rangle, a) - \widehat{Q}(s_0\langle H\rangle, a)\right| > \delta H$$

By setting $\delta = \frac{\Delta}{2H}$, we obtain that the error probability is bounded by

$$(\min(B,C)\cdot K)^H e^{-\frac{\Delta^2 C}{2H^4}}$$

The proof concludes by noting that the maximal loss for choosing non-optimal action at the root node $s_0\langle H\rangle$ is H.

References

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

Balla, R., & Fern, A. (2009). UCT for tactical assault planning in real-time strategy games. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 40–45.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Bjarnason, R., Fern, A., & Tadepalli, P. (2009). Lower bounding Klondike Solitaire with Monte-Carlo planning. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*.

Bonet, B., & Geffner, H. (2012). Action selection for MDPs: Anytime ao* vs. uct. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*.

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, *11*, 1–94.

Browne, C., Powley, E. J., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., & Colton, S. (2012). A survey of Monte-Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 1–43.

Bubeck, S., & Munos, R. (2010). Open loop optimistic planning. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, pp. 477–489.

Bubeck, S., Munos, R., & Stoltz, G. (2011). Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, *412*(19), 1832–1852.

Busoniu, L., & Munos, R. (2012). Optimistic planning for Markov decision processes. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, No. 22 in Journal of Machine Learning Research - Proceedings Track, pp. 182–189.

Coquelin, P.-A., & Munos, R. (2007). Bandit algorithms for tree search. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 67–74, Vancouver, BC, Canada.

Eyerich, P., Keller, T., & Helmert, M. (2010). High-quality policies for the Canadian Traveler's problem. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*.

Feldman, Z., & Domshlak, C. (2012). Simple regret optimization in online planning for markov decision processes. *CoRR*, `arXiv:1206.3382v2 [cs.AI]`.

Feldman, Z., & Domshlak, C. (2013). Monte-Carlo planning: Theoretically fast convergence meets practical efficiency. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 212–221.

Geffner, H., & Bonet, B. (2013). *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

Gelly, S., & Silver, D. (2011). Monte-Carlo tree search and rapid action value estimation in computer Go. *Artificial Intelligence*, *175*(11), 1856–1875.

Guestrin, C., Koller, D., Parr, R., & Venkataraman, S. (2003). Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, *19*, 399–468.

Hay, N., Shimony, S. E., Tolpin, D., & Russell, S. (2012). Selecting computations: Theory and applications. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence (UAI)*.

Kearns, M. J., Mansour, Y., & Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, *49*(2-3), 193–208.

Keller, T., & Eyerich, P. (2012). Probabilistic planning based on UCT. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS)*.

Keller, T., & Helmert, M. (2013). Trial-based heuristic tree search for finite horizon MDPs. In *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 135–143.

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning (ECML)*, pp. 282–293, Berlin, Germany.

Kolobov, A., Mausam, & Weld, D. (2012). LRTDP vs. UCT for online probabilistic planning. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*.

Mausam, & Kolobov, A. (2012). *Planning with Markov Decision Processes: An AI Perspective*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

Péret, L., & Garcia, F. (2004). On-line search for solving Markov decision processes via heuristic sampling. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI)*, pp. 530–534, Valencia, Spain.

Puterman, M. (1994). *Markov Decision Processes*. Wiley, New York.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, *58*(5), 527535.

Smith, S. J., & Nau, D. S. (1994). An analysis of forward pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1386–1391.

Sturtevant, N. (2008). An analysis of UCT in multi-player games. In *Proceedings of the 6th International Conference on Computers and Games (CCG)*, p. 3749.

Tolpin, D., & Shimony, S. E. (2011). Doing better than UCT: Rational Monte Carlo sampling in trees. *CoRR*, `arXiv:1108.3711v1 [cs.AI]`.

Tolpin, D., & Shimony, S. E. (2012). MCTS based on simple regret. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*.

Walsh, T. J., Goschin, S., & Littman, M. L. (2010). Integrating sample-based planning and model-based reinforcement learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 612–617.

Zilberstein, S. (1993). *Operational Rationality through Compilation of Anytime Algorithms*. Ph.D. thesis, University of California at Berkeley.