Online Planning in MDPs: Rationality and Optimization

Zohar Feldman
Industrial Engineering & Management
Technion, Israel

Carmel Domshlak
Industrial Engineering & Management
Technion, Israel

June 18, 2012

Abstract

We consider online planning in Markov decision processes. An algorithm for this problem should explore the set of possible policies from the current state, and, when interrupted, recommend an action to follow based on the outcome of the exploration. The performance of such an algorithm is assessed in terms of its simple regret, that is the loss in performance resulting from choosing the recommended action instead of an optimal one, and/or in terms of probability that the recommended action is not an optimal one. The best guarantees provided by the state-of-the-art algorithms for reduction of these measures over time are only polynomial. We introduce a new algorithm, BRUE, that achieves over time *exponential* reduction of these two measures. The algorithm is based on a simple yet non-standard state-space sampling scheme in which different samples are dedicated to different objectives. Our preliminary empirical evaluation shows that BRUE not only provides superior performance guarantees, but is also very effective in practice and favorably compares to state-of-the-art.

1 INTRODUCTION

In the past couple of decades, Markov decision processes (MDPs) have become a standard framework for planning under uncertainty. An MDP $\langle S, A, Tr, R\rangle$ is defined by a set of states S, a set of actions A, a stochastic transition function $Tr : S \times A \times S \to [0, 1]$, and a reward function

$R : S \times A \times S \to \mathbb{R}$. The current state of the agent is fully observable, and the objective of the agent is to act so to maximize its accumulated reward. In the finite horizon setting that will be used for most of the paper, the reward is accumulated along H steps.

The desire to attack problems of increasing complexity has led researches to consider online planning in MDPs. In online planning, the decision process of the agent is focused on the next action to perform, rather than on computing a quality policy for the entire MDP. The agent is given a generative model which allows for simulated execution of all possible sequences of actions, from any state of the MDP. The decision process consists of a simulation-based planning, terminated either according to a predefined schedule or due to an external interrupt, and followed by a recommendation of action to perform at the current state. Once that action is applied in the real environment, it modifies the environment and the decision process is repeated from the new state to select the next action and so on.

A prominent approach to online planning in MDPs, successfully used also in non-deterministic and imperfect information games, is Monte-Carlo (MC) planning. Numerous MC planning algorithms have been proposed in the literature. The sparse sampling algorithm by Kearns, Mansour, and Ng [10] offered near-optimal action selection in time exponential in H but independent of the state space size. Several other works, closer to our focus here, introduced interruptable algorithms that are designed to provide convergence to the best action if enough time is given, and small performance loss if the algorithm is stopped prematurely [18, 13, 12, 6, 5, 15, 19]. The quality of the action a recommended for state s with H steps-to-go is assessed in terms of the probability that a is sub-optimal, and in terms of the (closely related) *simple regret* $\Delta_H[s, a]$. The latter is the performance loss resulting from taking a and then following an optimal policy π^* for the remaining $H - 1$ steps, instead of following π^* from the beginning. That is,

$$\Delta_H[s, a] = Q_H(s, \pi^*(s, H)) - Q_H(s, a),$$

where

$$Q_H(s, a) = \mathbb{E}_{s'} \left[R(s, a, s') + Q_{H-1}(s', \pi^*(s', H - 1)) \right].$$

While empirical attractiveness of alternative MC planning algorithms depends on specifics of the problem in hand, only some of these algorithms provide formal guarantees on their expected performance improvement over time. In particular, none of the MC planning algorithms in use these days breaks the barrier of the worst-case polynomial reduction of error probability, and thus they at best guarantee polynomial simple regret over time.

MCT: [input: $\langle S, A, Tr, R\rangle$; $s_0 \in S$]
search tree $\mathcal{T} \leftarrow$ root node s_0
while time permits:
    $\rho \leftarrow$ probe-S$(s_0, \mathcal{T})$
    $\mathcal{T} \leftarrow$ expand-tree$(\mathcal{T}, \rho)$
    update-statistics$(\mathcal{T}, \rho)$
return recommend-action$(s_0, \mathcal{T})$

Figure 1: High-level scheme for regular Monte-Carlo tree sampling.

This is precisely the contribution of this paper: We introduce an MC planning algorithm for MDPs, BRUE, that achieves over time *exponential reduction of both the error probability and simple regret*. The key in this simple to implement and computationally efficient algorithm is decoupling between certain contradicting objectives that should be addressed by the sampling process for online MDP planning. Our preliminary empirical evaluation on a standard benchmark for comparison between MC planning algorithms shows that BRUE not only provides superior performance guarantees, but is also very effective in practice.

2 MONTE-CARLO PLANNING

A general scheme for Monte-Carlo planning, MCT, that gives rise to various specific algorithms for online MDP planning, is depicted in Figure 1. Starting with the current state s_0, MCT performs an iterative construction of a tree rooted at s_0. At each iteration, MCT issues a probe from s_0, expands the tree based on the outcome of the probe, and updates information stored at the nodes of the tree. Once the simulation phase is over, MCT issues a recommendation of action to perform in s_0, and this is based on the information collected at the nodes of the tree. For compatibility of the notation with prior literature, in what follows we refer to the tree nodes via the states associated with these nodes. Note that, 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 MCT forms a DAG over nodes $(s, h) \in S \times \{0, 1, \ldots, H\}$. By $A(s) \subseteq A$ in what follows we refer to the subset of actions applicable in state s.

Numerous concrete instances of MCT have been proposed, with UCT [12]

probably being the most popular such algorithm these days [8, 17, 3, 2, 7, 11]. To give a concrete sense of MCT's components, as well as to ground some intuitions discussed later on, below we describe the specific setting of MCT corresponding to the core UCT algorithm:

- probe-S: The probes $\rho = \langle s_0, a_1, s_1, \ldots, a_k, s_k \rangle$ are all issued from the root node s_0. The probe ends either when reached a sink state, that is, $A(s_k) = \emptyset$, or when $k = H$. Each node/action pair (s, a) is associated with a counter $n(s, a)$ and a value accumulator $\widehat{Q}(s, a)$; both $n(s, a)$ and $\widehat{Q}(s, a)$ are initialized to 0, and then updated by the update-statistics procedure. Given s_i, the next-on-the-probe action a_{i+1} is determined by the deterministic UCB1 policy [1], originally proposed for optimal cumulative regret minimization in stochastic multi-armed bandit (MAB) problems [14]: If $n(s_i, a) > 0$ for all $a \in A(s_i)$, then

$$a_{i+1} = \operatorname*{argmax}_a \left[\widehat{Q}(s_i, a) + c\sqrt{\frac{\log n(s_i)}{n(s_i, a)}} \right], \tag{1}$$

  where $n(s) = \sum_a n(s, a)$. Otherwise, a_{i+1} is selected uniformly at random from actions $a \in A(s_i)$ with $n(s_i, a) = 0$. In both cases, s_{i+1} is then sampled according to the conditional probability $\mathbb{P}(S|s_i, a_{i+1})$, induced by the transition function Tr.

- expand-tree:
  Each probe $\rho = \langle s_0, a_1, s_1, \ldots, a_k, s_k \rangle$ induces a state trace $\langle s_0, s_1, \ldots, s_i \rangle$ inside $\mathcal{T}$, as well as a state trace $\langle s_{i+1}, \ldots, s_k \rangle$ outside of $\mathcal{T}$. In principle, $\mathcal{T}$ can be expanded with any prefix of $\langle s_{i+1}, \ldots, s_k \rangle$; a practically popular choice appears to be expanding $\mathcal{T}$ with only the upper-most node s_{i+1}. (If $\mathcal{T}$ is constructed as a DAG, it is expanded with the first node along ρ that leaves $\mathcal{T}$.)

- update-statistics: For each node s_i along ρ that is now part of the expanded tree $\mathcal{T}$, the counter $n(s_i, a_{i+1})$ is incremented and the estimated Q-value is updated as

$$\widehat{Q}(s_i, a_{i+1}) \leftarrow \widehat{Q}(s_i, a_{i+1}) + \frac{R_i - \widehat{Q}(s_i, a_{i+1})}{n(s_i, a_{i+1})}, \tag{2}$$

  where $R_i = \sum_{j=i}^{k-1} R(s_j, a_{j+1}, s_{j+1})$.

- recommend-action: Interestingly, the action recommendation protocol of UCT was never properly specified, and different applications of UCT

adopt different decision rules, including maximization of the estimated Q-value, of the augmented estimated Q-value as in Eq. 1, of the number of times the action was selected during the simulation, as well as randomized protocols based on the information collected at the root.

The key property of MCT-based algorithms is that their exploration of the search space is obtained by considering a hierarchy of forecasters, each minimizing its own *cumulative* regret. Each such pseudo-agent forecaster corresponds to a state/steps-to-go pair (s, h). In that respect, according to Theorem 6 of Kocsis and Szepesvári [12], UCT achieves the optimal logarithmic cumulative regret. However, the cumulative regret does not seem to be the right way to base MC planning on, and this is because the rewards "collected" at the simulation phase are fictitious. In contrast, the same Theorem 6 of Kocsis and Szepesvári [12] claims only a polynomial-rate reduction of the probability of choosing a non-optimal action, and the recent results by Bubeck, Munos, and Stoltz [4] on simple regret minimization in MABs with stochastic rewards imply that UCT achieves only polynomial reduction of the simple regret over time. Some attempts have recently been made to adapt UCT, and MCT-based planning in general, to optimizing decisions in online MDP planning [19, 9]. As we show later on, some of these attempts were actually very successful empirically. However, to the best of our knowledge, none of them breaks the barrier for the formal performance guarantees provided by UCT.

3 THE BRUE ALGORITHM

The work of Bubeck et al. [4] was probably the first systematic attempt to analyze pure exploration in MABs, showing that the minimal simple regret in MAB can increase as the bound on the cumulative regret is getting smaller. The analysis of Bubeck et al. led us to suspect that the major deficiency of the current MC planning algorithms is in their very reliance on the MCT scheme, and to look for alternative search schemes that would better suit simple regret optimization in MDP planning. As a result, here we introduce a novel scheme for MC planning in MDPs, MCTer (see Figure 2), and then describe a concrete instance of this scheme that (1) guarantees that the probability of recommending non-optimal arm convergences to zero at exponential rate, and (2) achieves over time exponential rate of the simple regret reduction.[1]

[1]Note that we are *not* claiming that no instance of MCT can achieve such guarantees, but only that so far no such instance has been discovered. This leaves a very interesting

```
MCTer [input: ⟨S, A, Tr, R⟩; s_0 ∈ S]
search tree T ← root node s_0
while time permits:
    for h = H down to 1:
        ρ ← probe-S(s_0, T, h)
        T ← expand-tree(T, ρ)
        let ρ = ⟨s_0, a_1, s_1, ..., s_{h-1}, a_h, s_h⟩
        ρ̄ ← probe-R(s_h, T, H − h)
        update-statistics(T, ρ̄)
return recommend-action(s_0, T)
```

Figure 2: Monte-Carlo tree sampling with separation between exploration-oriented and recommendation-oriented samples.

Similarly to MCT, MCTer iteratively constructs a tree rooted at the current state s_0, starting each iteration with issuing a probe from s_0 and expanding the tree based on the outcome ρ of the probe. The probes ρ are issued to varying depth in a round-robin fashion, from the full depth of H to just a basic lookahead of 1. Importantly, instead of updating the statistics stored at the nodes based on ρ and proceeding with the next iteration as in MCT, a new probe $\overline{\rho}$ is issued from the end state of ρ, and the information at the tree nodes is updated only after this "complementary probe" $\overline{\rho}$. Importantly, the two type of probes can be generated according to different strategies, dubbed in Figure 2 as probe-S and probe-R.

The intuition behind what we try to achieve with the non-standard flow of MCTer is as follows. As we already mentioned, the origin of the standard MCT scheme, as well as of its various instantiations, is in online optimization in the context of various MAB problems. Considering each state/steps-to-go pair (s, h) as a pseudo-agent, the sole task of the root pseudo-agent (s_0, H) is to minimize its own simple regret in a stochastic MAB setting induced by the applicable actions $A(s_0)$. Therefore, if an oracle would provide (s_0, H) with an optimal action $\pi^*(s_0, H)$, then no further planning will be needed until after the execution of $\pi^*(s_0, H)$. However, the task characteristics of (s_0, H) is an exception rather than a rule. Suppose that an oracle provides us with optimal actions for *all* pseudo-agents (s, h) *but* (s_0, H). Despite the richness of this information, (s_0, H) in some sense remains as clueless as it was before. To choose between the alternatives $A(s_0)$, (s_0, H) needs at least

question for future work.

relative information about the expected value of these alternatives. Hence, each non-root pseudo-agent (s, h) is devoted to two tasks: (1) identifying an optimal action $\pi^*(s, h)$, and (2) estimating the actual value of that action, because this information is needed by the *predecessor(s)* of (s, h) in $\mathcal{T}$.

This is the key point that MCTer aims at targeting differently from MCT. While both schemes incrementally collect information about the search space by sampling it, they differ in the roles that different samples play in that process. In MCT, each probe ρ issued at (s, h) is a priori devoted *both* to increasing the confidence in that some current candidate $a^\dagger$ for $\pi^*(s, h)$ is indeed $\pi^*(s, h)$, as well as to improving the estimate of $Q_h(s, a^\dagger)$, as if assuming that $\pi^*(s, h) = a^\dagger$. Such an overloading of the probes is unavoidable in the "learning while acting" setup of reinforcement learning (RL) where agents should naturally care about their cumulative performance. However, while the second objective of (s, h) in online planning does prescribe (s, h) to act as to maximally *exploit* its current knowledge, this similarity to RL is somewhat misleading as RL-style exploitation *per se* is irrelevant in MC planning.

In principle, nothing requires our sampling mechanism to balance between the two *contradicting* objectives at the level of individual probes, the way instances of MCT are forced to do. In MCTer, the two roles are fulfilled by different sets of probes: the probes issued by probe-S aim at exploration while the probes issued by probe-R aim at improving the value estimates for the current candidates for π^*. In particular, this separation allows us to introduce a specific MCTer instance, BRUE,[2] that is tailored to simple regret minimization. Inspired by the positive results of Bubeck et al. [4] for MABs, the BRUE setting of MCTer is as follows:

- probe-S: The probes $\rho = \langle s_0, a_1, \ldots, s_{h-1}, a_h, s_h \rangle$ are all issued from the root node s_0, with actions along the probe being selected at random according to *uniform* distribution. Each node/action pair (s, a) is associated with $n(s, a)$ and $\widehat{Q}(s, a)$ as in UCT; while counters $n(s, a)$ are initialized to 0, value accumulators $\widehat{Q}(s, a)$ are schematically initialized to $-\infty$.

- expand-tree: $\mathcal{T}$ is expanded with the suffix of state sequence $s_1, \ldots, s_{h-1}$ that is new to $\mathcal{T}$.

- probe-R: In the complementary probe $\overline{\rho}$ issued at node s_h, $\overline{\rho} = \langle s_h = \overline{s}_1, \overline{a}_2, \overline{s}_2, \ldots, \overline{a}_k, \overline{s}_k \rangle$, each action $\overline{a}_i$ is sampled uniformly at random, but only among the actions $a \in A(\overline{s}_{i-1})$ that *maximize* $\widehat{Q}(\overline{s}_i, a)$.

[2] Short for **B**est **R**ecommendation with **U**niform **E**xploration.

- update-statistics: The complementary probe $\overline{\rho}$ as above is used to update statistics on the state/action pair (s_{h-1}, a_h) according to Eq. 2, but now with $R_i = \sum_{j=1}^{k-1} R(\overline{s}_j, \overline{a}_{j+1}, \overline{s}_{j+1})$. Note that the information obtained by $\overline{\rho}$ is *not* pushed further up the probe ρ. While that may appear wasteful and even counterintuitive, this locality of update is required to satisfy the formal guarantees of BRUE discussed later on.

- recommend-action: The action recommended by BRUE is chosen uniformly at random among the actions a maximizing $\widehat{Q}(s_0, a)$.

For the sake of simplicity, in our formal analysis we assume uniqueness of the optimal policy π^*; that is, at each state s and each number h of steps-to-go, there is a single optimal action, and it is $\pi^*(s, h)$. Likewise, let π_n^{B} be the (possibly stochastic) policy induced by the value accumulators $\widehat{Q}$ after n iterations of BRUE: denoting by $\mathcal{T}_n$ the graph obtained by BRUE after n iterations, and by $\widehat{Q}_h(s, a)$ the accumulated value $\widehat{Q}(s, a)$ for s at depth $H - h$, for all state/steps-to-go pairs $(s, h) \in \mathcal{T}_n$, $\pi_n^{\mathrm{B}}(s, h)$ is a randomized strategy, uniformly choosing among actions a maximizing $\widehat{Q}_h(s, a)$. We also use some auxiliary notation:

K $= \max_{s \in S} |A(s)|$, i.e., the maximal number of actions per state.

p $= \min_{s,a,s' : Tr(s,a,s') > 0} Tr(s, a, s')$, i.e., the likelihood of the least likely (but still possible) outcome of an action in our problem.

d $= \min_{s,a} \Delta_1[s, a]$, i.e., the smallest difference between the value of the optimal and a second-best action at a state with just one step-to-go.

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. For any $1 \leq h \leq H$, there exist parameters $c_h, c'_h, c''_h < \infty$, dependent of p, d and h, but independent of s_0, $|S|$, and the number of* BRUE *iterations n, such that, for each state s reachable from s_0 in $H - h$ steps, for all $n \geq c''_h$, we have*

- *Simple Regret*
$$\mathbb{E}\Delta_h[s, \pi_n^B(s, h)] \leq c_h e^{-c'_h n}, \tag{3}$$

- *Error Probability*
$$\mathbb{P}\left\{\pi_n^B(s, h) \neq \pi^*(s, h)\right\} \leq \frac{c_h}{h} e^{-c'_h n}. \tag{4}$$

Corollary 1 *The simple regret of the action $\pi_n^B(s_0, H)$, recommended by* BRUE *after $n > c_1$ iterations, is bounded by $c_2 \cdot \exp(-n \cdot c_3)$, where $c_i = f_i(p, d, H) < \infty$.*

In rest of this section we prove Theorem 1. To shorten the equations, when considering in what follows a state s with h steps-to-go, by a^* we denote the optimal action $\pi^*(s, h)$. When considering in that context an action $a \in A(s)$, by Q we denote $Q_h(s, a)$, and by $\widehat{Q}$ and $n(a)$ we denote the respective value accumulator $\widehat{Q}_h(s, a)$ and counter $n(s, a)$; Q^* and $\widehat{Q}^*$ denote $Q_h(s, a^*)$ and $\widehat{Q}_h(s, a^*)$, respectively. Likewise, by p_h we denote the minimal probability of reaching a (reachable this way) state with still h steps-to-go on a uniform sample from s_0.

The proof is by induction on h. Recall that the recommended action $\pi_n^{\mathrm{B}}(s, h)$ is an action maximizing $\widehat{Q}_h(s, a)$. Starting with proving the induction basis for $h = 1$, note that the probability of choosing an action $a \neq a^*$ is bounded by

$$\mathbb{P}\left\{\widehat{Q} \geq \widehat{Q}^*\right\} \leq \mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_1[s,a]}{2}\right\} + \mathbb{P}\left\{\widehat{Q} \geq Q + \frac{\Delta_1[s,a]}{2}\right\} \tag{5}$$

The analysis of the two summation terms in Eq. 5 is identical, and thus we detail it here only for the first term.

$$\begin{aligned} &\mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_1[s,a]}{2}\right\} \leq \\ &\qquad \mathbb{P}\left\{n(a^*) \leq n_0\right\} + \mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_1[s,a]}{2}, n(a^*) > n_0\right\} \end{aligned} \tag{6}$$

To bound the first term of this summation, we note that, for each search node (s, h) and each action $a \in A(s)$, $n(a)$ is a sum of n independent Bernouli trials with success probability $p_h \geq \frac{1}{K}(\frac{p}{K})^{H-h}$, and $\mathbb{E}[n(a)] \geq np_h$. By choosing $n_0 = n\frac{p_1}{2}$ and employing Chernoff-Hoefdding bound, we obtain

$$\begin{aligned} \mathbb{P}\left\{n(a^*) \leq n_0\right\} &= \mathbb{P}\left\{\frac{n(a^*)}{n} \leq \frac{\mathbb{E}[n(a^*)]}{n} - \left(\frac{\mathbb{E}[n(a^*)]}{n} - \frac{n_0}{n}\right)\right\} \\ &\leq \exp\left(-2n\left(\frac{p^{(H-1)}}{2K^H}\right)^2\right) = \exp\left(-n\frac{p_1^2}{2}\right). \end{aligned} \tag{7}$$

The second term in Eq. 6 can be bounded as:

$$
\begin{aligned}
&\mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_1[s,a]}{2}, n(a^*) > n_0\right\} \\
&= \sum_{t=n_0+1}^{n} \mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_1[s,a]}{2}, n(a^*) = t\right\} \\
&\leq \sum_{t=n_0+1}^{n} \mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_1[s,a]}{2} \;\middle|\; n(a^*) = t\right\} \mathbb{P}\left\{n(a^*) = t\right\} \\
&= \sum_{t=n_0+1}^{n} \mathbb{P}\left\{\widehat{Q}^* \leq \mathbb{E}\left[\widehat{Q}^*\right] - \frac{\Delta_1[s,a]}{2} \;\middle|\; n(a^*) = t\right\} \mathbb{P}\left\{n(a^*) = t\right\} \\
&\quad \text{since } \mathbb{E}\left[\widehat{Q}^*\right] = Q^* \\
&\leq \sum_{t=n_0+1}^{n} \exp\left(-2t\frac{\Delta_1[s,a]^2}{4}\right) \mathbb{P}\left\{n(a^*) = t\right\} \\
&\quad \text{by Chernoff-Hoefdding bound} \\
&\leq \quad e^{-2n_0 \frac{\Delta_1[s,a]^2}{4}} \sum_{t=n_0+1}^{n} \mathbb{P}\left\{n(a^*) = t\right\} \\
&\leq \quad e^{-n\frac{d^2 p_1}{4}}. \qquad (8)
\end{aligned}
$$

Putting together Eqs. 5-8, we obtain $\mathbb{P}\left\{\widehat{Q} \geq \widehat{Q}^*\right\} \leq 4\exp\left(-n\frac{d^2 p_1^2}{4}\right)$, and thus, for all $n \geq 1$,

$$
\mathbb{E}\Delta_1[s, \pi_n^{\mathrm{B}}(s,1)] \leq \sum_{a \neq a^*} \Delta_1[s,a] \cdot \mathbb{P}\left\{\widehat{Q} \geq \widehat{Q}^*\right\} \leq 4Ke^{-n\frac{d^2 p_1^2}{4}} \qquad (9)
$$

and

$$
\mathbb{P}\left\{\pi_n^{\mathrm{B}}(s,1) \neq \pi^*(s,1)\right\} \leq \sum_{a \neq a^*} \mathbb{P}\left\{\widehat{Q} \geq \widehat{Q}^*\right\} \leq 4Ke^{-n\frac{d^2 p_1^2}{4}} \qquad (10)
$$

Now, assuming the claim holds for $h \geq 1$, we prove it for $h+1$. The probability of choosing a sub-optimal action $a \neq a^*$ when still $h+1$ steps-to-go is bounded similarly to Eq. 5, with $\Delta_1[s,a]$ replaced by $\Delta_{h+1}[s,a]$. Similarly, we obtain

$$
\begin{aligned}
&\mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_{h+1}[s,a]}{2}\right\} \leq \\
&\quad \mathbb{P}\left\{n(a^*) \leq n_0\right\} + \mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_{h+1}[s,a]}{2}, n(a^*) > n_0\right\}
\end{aligned} \qquad (11)
$$

and

$$\mathbb{P}\left\{n(a^*) \leq n_0\right\} \leq \exp\left(-n\frac{p_{h+1}^2}{2}\right). \tag{12}$$

However, bounding the second term of summation in Eq. 11 differs from this for $h = 1$ in Eq. 8 for two reasons.

(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, h).

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

We now show that these deficiencies of $h > 1$ can still be overcome. In what follows, we first tackle the issues F1-F2 by a different bounding of the quantity

$$\mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_{h+1}[s,a]}{2} \;\middle|\; n(a^*) = t\right\},$$

and then use the outcome of this analysis to bound the second summand of Eq. 11.

Lemma 1 (Expected accumulated rewards) *Considering the policy π_n^B induced by n iterations*[3] *of* BRUE *on an MDP $\langle S, A, Tr, R\rangle$, for each state $s \in S$, each action $a \in A(s)$, and each number of steps-to-go $h \in \{1, \ldots, H\}$, let $X = R(s, a, s_1) + \sum_{i=1}^{h-1} R\left(s_i, \pi_n^B(s_i, h-i), s_{i+1}\right)$ be a random variable corresponding to the reward accumulated by taking a at s, and then following π_n^B for the rest $h-1$ steps. Then $\delta_{n,h}(s,a) \triangleq Q(s,a) - \mathbb{E}[X]$ is*

$$\delta_{n,h}(s,a) = \sum_{i=1}^{h-1} \mathbb{E}\Delta_{h-i}[s_i, \pi_n^B(s_i, h-i)]. \tag{13}$$

Proof: For any state/steps-to-go pair $(s,h) \in S \times \{1, \ldots, H\}$, we have

$$\begin{aligned} &\mathbb{E}_{\mathrm{B},s'}\left[R\left(s, \pi_n^{\mathrm{B}}(s,h), s'\right)\right] = \\ &\quad \mathbb{E}_{\mathrm{B}}\left[Q(s, \pi_n^{\mathrm{B}}(s,h))\right] - \mathbb{E}_{\mathrm{B},s'}\left[Q(s', \pi^*(s', h-1))\right]. \end{aligned} \tag{14}$$

[3]Iteration = full run of the most-inner loop of BRUE.

Using that, we obtain a telescopic series that yields

$$\begin{aligned}
\mathbb{E}\left[X\right] &= \mathbb{E}_{\mathrm{B},s_1:s_h}\left[R\left(s,a,s_1\right) + \sum_{i=1}^{h-1} R\left(s_i, \pi_n^{\mathrm{B}}(s_i, h-i), s_{i+1}\right)\right] \\
&= Q(s,a) - \mathbb{E}_{s_1}\left[Q(s_1, \pi^*(s_1, h-1))\right] + \\
&\quad \sum_{i=1}^{h-1} \mathbb{E}_{\mathrm{B},s_1:s_i}\left[Q(s_i, \pi_n^{\mathrm{B}}(s_i, h-i))\right] - \\
&\quad \sum_{i=1}^{h-1} \mathbb{E}_{\mathrm{B},s_1:s_{i+1}}\left[Q(s_{i+1}, \pi^*(s_{i+1}, h-i-1))\right] \\
&= Q(s,a) - \sum_{i=1}^{h-1} \mathbb{E}_{\mathrm{B},s_1:s_i}\left[\Delta_{h-i}[s_i, \pi_n^{\mathrm{B}}(s, h-i)]\right].
\end{aligned} \tag{15}$$

■

In what follows, we refer to the supremum of $\delta_{n,h}(s,a)$ from Lemma 1 as $\delta_{n,h} \triangleq \max_{s,a}\{\delta_{n,h}(s,a)\}$.

With Lemma 1 in hand, let $\{X_t\}_{t=1}^{n(a^*)}$ be the summands of the value accumulator $\widehat{Q}^* = \widehat{Q}_{h+1}(s,a^*)$. Approaching (F1) first, we exploit the fact that the entire bias in our estimate of Q^* is due to the erroneous recommendations of π^{B} at the successors of s, and the damage of these erroneous recommendations is fully captured by the simple regret of using π^{B} at the *immediate* successors of s. When bounding $Q^* - \mathbb{E}X_t$, we note that this difference is largest when each X_t is sampled at iteration t, that is, each X_t is acquired at the earliest possible (and thus least informed) stage of the algorithm. Hence, $Q^* - \mathbb{E}X_t \leq \delta_{t,h+1}$, and this allows us to bound the influence of the bias as follows.

$$\begin{aligned}
&\mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_{h+1}[s,a]}{2} \;\middle|\; n(a^*) = t\right\} \\
&\quad = \mathbb{P}\left\{\widehat{Q}^* \leq \mathbb{E}\widehat{Q}^* - \left(\frac{\Delta_{h+1}[s,a]}{2} - \left(Q^* - \mathbb{E}\widehat{Q}^*\right)\right) \;\middle|\; n(a^*) = t\right\} \\
&\quad \leq \mathbb{P}\left\{\widehat{Q}^* \leq \mathbb{E}\widehat{Q}^* - \left(\frac{\Delta_{h+1}[s,a]}{2} - \frac{1}{t}\sum_{\tau=1}^{t}\delta_{\tau,h+1}\right) \;\middle|\; n(a^*) = t\right\} \\
&\quad \text{since } Q^* - \mathbb{E}\widehat{Q}^* = \frac{1}{t}\sum_{\tau=1}^{t}\left(Q^* - \mathbb{E}X_\tau\right) \leq \frac{1}{t}\sum_{\tau=1}^{t}\delta_{\tau,h+1}
\end{aligned} \tag{16}$$

We now proceed with (F2), bounding the influence of the $\widehat{Q}^*$ summands dependency. At high level, we achieve that by modifying the Chernoff-Hoefdding bound for independent random variables to certain sequences of dependent random variables $\{Y_t\}$ such that $Y_t|Y_{t-1},\ldots,Y_1$ concentrates around its expectation with a probability that approaches 1 at rate exponential in t.

We begin with establishing some auxiliary notions and properties. Let A_t be the event in which X_t is sampled along the optimal actions in each of the h subsequent nodes[4]. We note that

$$\mathbb{P}\left\{\neg A_t\right\} \leq \sum_{i=1}^{h} \mathbb{P}\left\{\pi_t^{\mathrm{B}}(s_i,i) \neq \pi^*(s_i,i)\right\}. \tag{17}$$

Denoting $c_{p,h} = \sum_{i=1}^{h} \frac{c_i}{i}$ and $c_{\delta,h} = \sum_{i=1}^{h} c_i$, from (stemming from induction hypothesis) Eq. 10, monotonic decrease of c_i' in i, and monotonic increase of c_i'' in i, for all $t \geq c_h''$ we have

$$\mathbb{P}\left\{\neg A_t\right\} \leq \sum_{i=1}^{h} \frac{c_i}{i} e^{-c_i' t} \leq c_{p,h} e^{-c_h' t}, \tag{18}$$

and

$$\delta_{t,h+1} \leq \sum_{i=1}^{h} c_i e^{-c_i' t} \leq c_{\delta,h} e^{-c_h' t}. \tag{19}$$

Now, for all $\omega \in A_t$, $\mathbb{E}\left[X_t \mid X_1,\ldots,X_{t-1}\right](\omega) = Q^*$, and thus $\left[Q^* - X_t \mid X_1,\ldots,X_{t-1}\right](\omega)$ is a random variable with expectation 0, and support $[Q^* - (h+1), Q^*]$. Hence, for any $\lambda \geq 0$,

$$\begin{aligned}
&\mathbb{E}\left[e^{\lambda(\mathbb{E}X_t - X_t)} \;\middle|\; X_1,\ldots,X_{t-1}\right] \\
&\quad = e^{\lambda(\mathbb{E}X_t - Q^*)} \mathbb{E}\left[e^{\lambda(Q^* - X_t)} \;\middle|\; X_1,\ldots,X_{t-1}\right] \\
&\quad \leq e^{-\lambda\delta_{t,h+1}} \mathbb{E}\left[e^{\lambda(Q^* - X_t)} \;\middle|\; X_1,\ldots,X_{t-1}\right] \\
&\quad \leq e^{-\lambda\delta_{t,h+1} + \frac{\lambda^2 (h+1)^2}{8}} \\
&\quad \text{using Fact 1 below.}
\end{aligned} \tag{20}$$

[4]It is easy to see that $A_t \in \sigma\left(X_1,\ldots,X_{t-1}\right)$

Fact 1 *Let Z be a random variable satisfying $a \leq Z \leq b$ and $\mathbb{E}[Z] = 0$. Then, $\mathbb{E}[\exp(\lambda Z)] \leq \exp(\frac{(b-a)^2\lambda^2}{8})$ for any $\lambda \in \mathbb{R}$.*

Proceeding now with developing Eq. 16, let $\alpha = \frac{\Delta_{h+1}[s,a]}{2} - \frac{1}{t}\sum_{\tau=1}^{t}\delta_{\tau,h}$. If $\alpha > 0$, then for all $\lambda \geq 0$, from Markov inequality it holds that

$$\begin{aligned}(16) &= \mathbb{P}\left\{\widehat{Q}^* \leq \mathbb{E}\widehat{Q}^* - \alpha \;\middle|\; n(a^*) = t\right\} \\ &= \mathbb{P}\left\{t\widehat{Q}^* \leq t\mathbb{E}\widehat{Q}^* - t\alpha \;\middle|\; n(a^*) = t\right\} \\ &\leq e^{-\lambda t\alpha}\mathbb{E}\left[e^{\lambda\sum_{i=1}^{t}(\mathbb{E}X_i - X_i)}\right]. \qquad (21)\end{aligned}$$

Focusing on the second multiplicand in Eq. 21,

$$\begin{aligned}&\mathbb{E}\left[e^{\lambda\sum_{i=1}^{t}(\mathbb{E}X_i - X_i)}\right] \\ &= \mathbb{E}_{A_t}\left[e^{\lambda\sum_{i=1}^{t}(\mathbb{E}X_i - X_i)}\right] + \mathbb{E}_{\neg A_t}\left[e^{\lambda\sum_{i=1}^{t}(\mathbb{E}X_i - X_i)}\right] \\ &\leq e^{-\lambda\delta_{t,h+1} + \frac{\lambda^2(h+1)^2}{8}}\mathbb{E}\left[e^{\lambda\sum_{i=1}^{t-1}(\mathbb{E}X_i - X_i)}\right] + \mathbb{P}\{\neg A_t\}\, e^{\lambda t Q^*} \\ &\leq e^{\frac{\lambda^2(h+1)^2}{8}}\mathbb{E}\left[e^{\lambda\sum_{i=1}^{t-1}(\mathbb{E}X_i - X_i)}\right] + c_{p,h}e^{t\left(\lambda(h+1) - c'_h\right)} \\ &\quad \text{by Eq. 18, and } Q^* \leq h+1 \\ &\leq e^{\frac{\lambda^2(h+1)^2}{8}\left(t - c''_h\right)}e^{\lambda c''_h(h+1)} \;+ \\ &\quad \sum_{\tau=c''_h+1}^{t} e^{\frac{\lambda^2(h+1)^2}{8}(t-\tau)}c_{p,h}e^{\tau\left(\lambda(h+1) - c'_h\right)} \qquad (22) \\ &\quad \text{by iterating until } c''_h \text{ along Eq. 24 as described below.}\end{aligned}$$

Considering the recursion

$$f(t) = \theta f(t-1) + g(t), \qquad (23)$$

it can be shown that, by iterating until $t = c$, Eq. 23 is equivalent to

$$f(t) = \theta^{t-c}f(c) + \sum_{\tau=c+1}^{t}\theta^{t-\tau}g(\tau). \qquad (24)$$

Given that, the last bound in Eq. 22 is obtained by setting $f(t) = \mathbb{E}\left[e^{\lambda\sum_{i=1}^{t}(\mathbb{E}X_i - X_i)}\right]$, $\theta = e^{\frac{\lambda^2(h+1)^2}{8}}$, and $g(t) = c_{p,h}e^{t\left(\lambda(h+1) - c'_h\right)}$.

Denoting now $\xi = \frac{\alpha c'_h}{(h+1)}$, and choosing $\lambda = \xi \frac{2}{(h+1)}$,

$$
\begin{aligned}
(22) &= e^{\xi^2 \frac{(t-c''_h)}{2}} e^{2\xi c''_h} + \sum_{\tau=c''_h+1}^{t} e^{\xi^2 \frac{(t-\tau)}{2}} c_{p,h} e^{\tau\left(2\xi - c'_h\right)} \\
&\leq e^{\xi^2 \frac{(t-c''_h)}{2}} e^{2\xi c''_h} + \sum_{\tau=c''_h+1}^{t} e^{\xi^2 \frac{(t-\tau)}{2}} c_{p,h} \\
&\quad \text{since, by definition of } \alpha, \text{ it holds that } \xi \leq \frac{c'_h}{2} \\
&= e^{\xi^2 \frac{(t-c''_h)}{2}} e^{2\xi c''_h} + c_{p,h} e^{\xi^2 \frac{t}{2}} \sum_{\tau=c''_h+1}^{t} e^{-\xi^2 \frac{\tau}{2}} \\
&\leq e^{\xi^2 \frac{(t-c''_h)}{2}} e^{2\xi c''_h} + c_{p,h} e^{\xi^2 \frac{t}{2}} \left(\frac{2}{\xi^2} e^{-\frac{\xi^2}{2} c''_h} \right) \\
&\quad \text{since } \sum_{t=c+1}^{\infty} e^{-kt} \leq \frac{1}{k} e^{-ck} \\
&\leq e^{\xi^2 \frac{t}{2}} e^{2\xi c''_h} + \frac{2 c_{p,h}}{\xi^2} e^{\xi^2 \frac{t}{2}}. \qquad (25)
\end{aligned}
$$

Using the bound provided by Eq. 25 for $\lambda = \xi \frac{2}{(h+1)}$ into Eq. 21, we obtain

$$
\begin{aligned}
&\mathbb{P}\left\{ \widehat{Q}^* \leq \mathbb{E}\widehat{Q}^* - \alpha \;\middle|\; n(a^*) = t \right\} \\
&\qquad \leq e^{-\frac{3\alpha^2 c'_h}{2(h+1)^2} t} \left(e^{\frac{2\alpha c'_h c''_h}{h+1}} + \frac{2 c_{p,h} \, (h+1)^2}{\alpha^2 c'^2_h} \right). \qquad (26)
\end{aligned}
$$

Since this bound is true only for $\alpha > 0$, it holds only starting from n such that $\frac{\Delta_{h+1}[s,a]}{2} > \frac{1}{t} \sum_{\tau=1}^{t} \delta_{\tau,h+1}$ for all $t \geq n_0 = \frac{n p_{h+1}}{2}$. Recalling Eq. 19, by induction hypothesis, for $t \geq c''_h$ it holds that

$$
\begin{aligned}
\sum_{\tau=1}^{\infty} \delta_{\tau,h+1} &= \sum_{\tau=1}^{c''_h} \delta_{\tau,h+1} + \sum_{\tau=c''_h+1}^{\infty} \delta_{\tau,h+1} \\
&\leq c''_h (h+1) + \sum_{\tau=c''_h+1}^{\infty} c_{\delta,h} e^{-c'_h \tau} \qquad (27) \\
&\leq c''_h (h+1) + \frac{c_{\delta,h}}{c'_h} e^{-c'_h c''_h} \quad \triangleq \quad \beta.
\end{aligned}
$$

Therefore, for $t \geq \frac{2}{d}\beta$, we have that $\alpha > 0$. Furthermore, to simplify dealing with α in the denominator in Eq. 26, for $t \geq \frac{4}{d}\beta$, we have $\alpha \geq \frac{\Delta_{h+1}[s,a]}{4}$. Hence, for all $n \geq c''_{h+1} = \frac{8}{dp_{h+1}}\beta$, we have that $\alpha \geq \frac{\Delta_{h+1}[s,a]}{4}$ for all $t > n_0$. Consequently, from Eq. 26, for all $n \geq c''_{h+1}$,

$$\begin{aligned}
&\mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_{h+1}[s,a]}{2} \;\middle|\; n(a^*) = t\right\} \\
&\leq e^{-\frac{3c'_h d^2}{8(h+1)^2}t} e^{\frac{3c'_h}{2(h+1)}\beta}\left(e^{\frac{2\alpha c'_h c''_h}{h+1}} + \frac{2c_{p,h}\,(h+1)^2}{\alpha^2 c'^2_h}\right) \\
&\text{since } -t\alpha^2 \leq -t\left(\frac{\Delta_{h+1}[s,a]^2}{4} - \Delta_{h+1}[s,a]\frac{\beta}{t}\right) \leq -\frac{td^2}{4} + (h+1)\beta \\
&\leq e^{-\frac{3c'_h d^2}{8(h+1)^2}t} e^{\frac{3c'_h}{2(h+1)}\beta}\left(e^{c'_h c''_h} + c_{p,h}\frac{32\,(h+1)^2}{d^2 c'^2_h}\right) \triangleq \gamma(t). \\
&\text{since } \frac{d}{4} \leq \alpha \leq \frac{h+1}{2}
\end{aligned}$$

This now provides us with the desired bound on the second summand in Eq. 11 as

$$\begin{aligned}
&\mathbb{P}\left\{\widehat{Q}^* \leq Q^* - \frac{\Delta_{h+1}[s,a]}{2}, n(a^*) > n_0\right\} \\
&\leq \sum_{t=n_0}^{n} \gamma(t)\cdot\mathbb{P}\left\{n(a^*) = t\right\} \;\leq\; \gamma(n_0)\sum_{t=n_0}^{n}\mathbb{P}\left\{n(a^*) = t\right\} \\
&\leq \gamma(n_0) = e^{-n\frac{3c'_h d^2 p_{h+1}}{16(h+1)^2}} e^{\frac{3c'_h}{2(h+1)}\beta}\left(e^{c'_h c''_h} + c_{p,h}\frac{32\,(h+1)^2}{d^2 c'^2_h}\right)
\end{aligned}$$

This finalizes the proof of induction hypothesis with

$$\begin{aligned}
c_{h+1} &= 2\left(1 + e^{\frac{3c'_h}{2(h+1)}\beta}\left(e^{c'_h c''_h} + c_{p,h}\frac{32\,(h+1)^2}{d^2 c'^2_h}\right)\right), \\
c'_{h+1} &= \frac{3c'_h d^2 p^2_{h+1}}{16\,(h+1)^2}, \\
c''_{h+1} &= \frac{8}{dp_{h+1}}\beta.
\end{aligned}$$

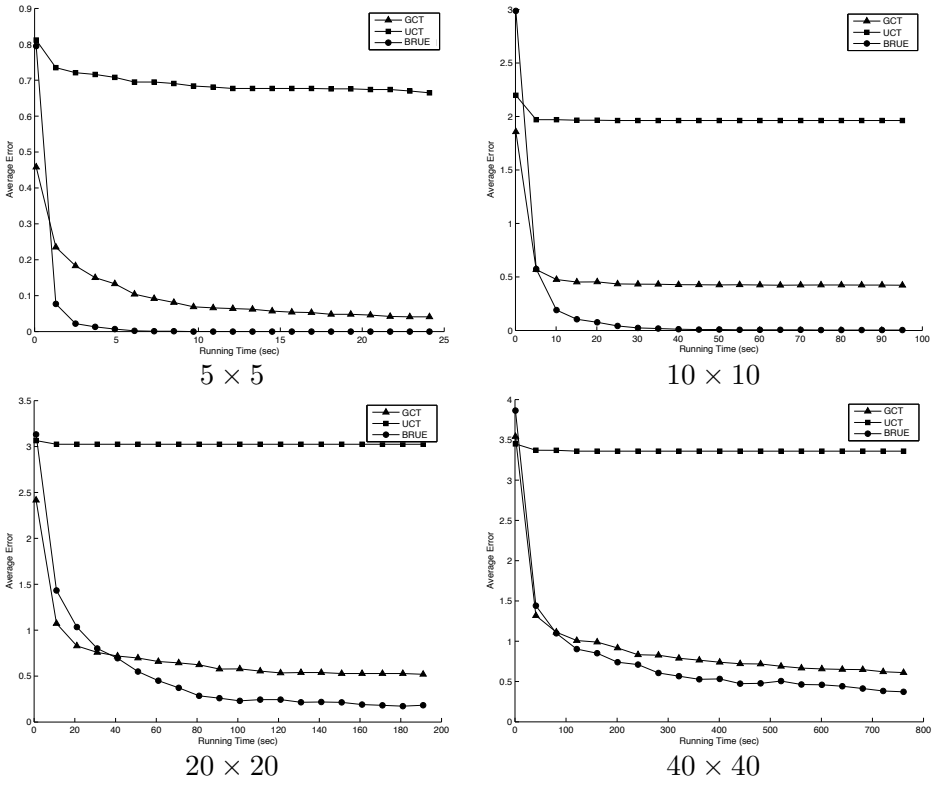
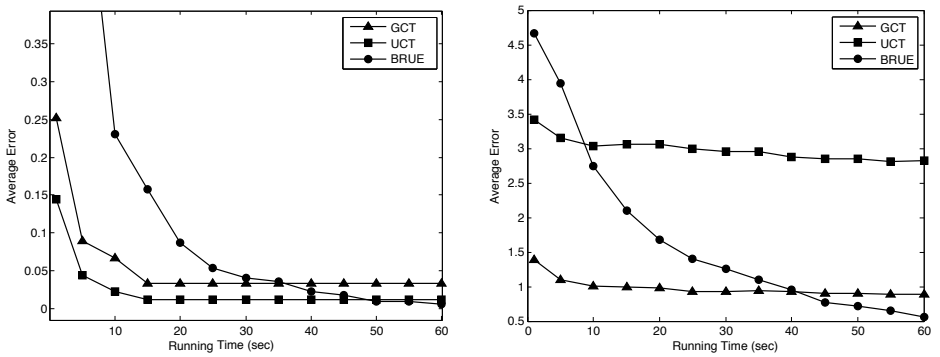


Figure 3: Empirical performance of BRUE, UCT, and GCT in terms of the average error on sailing domain problems on $n \times n$ grids with $n \in \{5, 10, 20, 40\}$.

4 EXPERIMENTAL EVALUATION

We have evaluated BRUE empirically on the MDP sailing domain [13] that was used in previous works for evaluating MC planning algorithms [13, 12, 19], as well as on random game trees used in the original empirical evaluation of UCT [12].

In 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 in as short time 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



$B = 6/D = 6$ $B = 2/D = 16$

Figure 4: Empirical performance of BRUE, UCT, and GCT in terms of the average error in the random game trees domain.

and moves fastest under 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 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 shall construct its tree. In the sailing domain, we chose H to be $4 \times n$, where n is the grid-size of the problem instance, as it is unlikely that the optimal path between any two locations on the grid will be larger than a complete encircling of the considered area. We note, however, that the recommendation-oriented samples $\bar{\rho}$ always end at a terminal state, similar to the rollouts issued by UCT and GCT.

Snapshots of the results for different grid sizes are shown in Figure 3. The comparison was made with two MCT-based algorithms: the UCT algorithm, and a recent modification of UCT, GCT, obtained from the former by replacing the UCB1 policy *at the root node* with the ϵ-greedy policy [19]. The motivation behind the design of GCT was to improve the empirical simple regret of UCT, and the results for GCT reported by [19] (and confirmed by our experiments here) are very impressive. All three algorithms were implemented within a single software infrastructure, with the parameters for UCT and GCT being set as in the previously reported evaluations on the sailing domain. Each algorithm was run on 1000 randomly chosen initial states s_0, and the performance of the algorithm was assessed in terms of the average error $Q(s_0, a) - V(s_0)$, that is, the difference between the true values of

the action a chosen by the algorithm and this of the optimal action $\pi^*(s_0)$. Consistently with the results reported by Tolpin and Shimony ([19]), GCT outperformed UCT by a very large margin, with the latter exhibiting very poor performance improvement over time even on the smallest, 5×5, grids. In turn, BRUE substantially outperformed GCT, with the improvement being consistent except for relatively short planning deadlines.

The above allows us to conclude that BRUE is not only attractive in formal terms of performance guarantees, but can also very effective in practice of online planning. Under the same parameter setting of UCT and GCT, we have also evaluated the three algorithms in a domain of random game trees that aims at 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. The rewards 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*. This simple game tree model is similar in spirit to many other game tree models used in previous work [12, 16], except for that the success/failure of the players in measured not on a ternary scale of win/lose/draw, but via the actual payoffs received by the players. We have ran some preliminary experiments with two different settings of the branching factor (B) and tree depths (D). As in the sailing domain, we compared the convergence rate obtained by BRUE, UCT and GCT. Figure 4 plots the average error rate for two configurations, $B = 6, D = 6$ and $B = 2, D = 16$, with the average in each setting obtained over 20 trees and 100 runs for each tree. The results here appear encouraging as well, with BRUE taking over the other two algorithms faster on the deeper trees.

5 SUMMARY AND DISCUSSION

We have introduced BRUE, a simple Monte-Carlo algorithm for online planning in MDPs that guarantees exponential-over-time reduction of the performance measures of interest, namely the simple regret and the probability

of erroneous action choice. This improves over previous algorithms such as UCT that guarantee only polynomial-over-time reduction of these measures. The algorithm has been formalized for finite horizon MDPs, and it was analyzed as such. However, our empirical evaluation shows that it performs well also on goal-driven MDPs and two-person games.

Our work leaves a few questions for future work. Considering γ-discounted MDPs with infinite horizon, a straightforward way to employ BRUE in that setting is to fix a horizon H, use the algorithm as it is, and derive guarantees on the aforementioned measures of interest by simply accounting for the additive gap of $\gamma^H R_{\max}/(1-\gamma)$ between the state/action values under horizon H and these under infinite horizon. However, this is not necessarily the best way to plan online for infinite-horizon MDPs, and thus this setting requires further introspection. Second, it is not unlikely that the state-space independent factors c_h, c'_h, and c''_h in the guarantees of BRUE can be substantially improved by employing more sophisticated combinations of exploration and recommendation probes, and currently we examine this issue. Finally, the core tree sampling scheme employed by BRUE differ from the more standard scheme employed in previous work. While this difference plays a critical role in establishing the formal guarantees of BRUE, it is still unclear whether that difference is *necessary* for establishing exponential-over-time reduction of the performance measures.

Acknowledgements

Work supported by and carried out at the Technion-Microsoft Electronic Commerce Research Center.

References

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

[2] R. Balla and A. Fern. UCT for tactical assault planning in real-time strategy games. In *IJCAI*, pages 40–45, 2009.

[3] R. Bjarnason, A. Fern, and P. Tadepalli. Lower bounding Klondike Solitaire with Monte-Carlo planning. In *ICAPS*, 2009.

[4] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theor. Comput. Sci.*, 412(19):1832–1852, 2011.

[5] T. Cazenave. Nested Monte-Carlo search. In *IJCAI*, pages 456–461, 2009.

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

[7] P. Eyerich, T. Keller, and M. Helmert. High-quality policies for the Canadian Traveler's problem. In *AAAI*, 2010.

[8] S. Gelly and D. Silver. Monte-Carlo tree search and rapid action value estimation in computer Go. *AIJ*, 175(11):1856–1875, 2011.

[9] N. Hay and S. J. Russell. Metareasoning for Monte Carlo tree search. Technical Report UCB/EECS-2011-119, EECS Department, University of California, Berkeley, Nov 2011.

[10] M. J. Kearns, Y. Mansour, and A. Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In *IJCAI*, pages 1324–1231, 1999.

[11] T. Keller and P. Eyerich. Probabilistic planning based on UCT. In *ICAPS*, 2012.

[12] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *ECML*, pages 282–293, 2006.

[13] L. Péret and F. Garcia. On-line search for solving Markov decision processes via heuristic sampling. In *ECAI*, pages 530–534, 2004.

[14] H. Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527535, 1952.

[15] C. D. Rosin. Nested rollout policy adaptation for Monte Carlo tree search. In *IJCAI*, pages 649–654, 2011.

[16] S. J. Smith and D. S. Nau. An analysis of forward pruning. In *AAAI*, pages 1386–1391, 1994.

[17] N. Sturtevant. An analysis of UCT in multi-player games. In *CCG*, page 3749, 2008.

[18] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

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