Exploiting Variance Information in Monte-Carlo Tree Search

Robert Lieck **Vien Ngo** **Marc Toussaint**
Machine Learning and Robotics Lab
University of Stuttgart
prename.surname@ipvs.uni-stuttgart.de

Abstract

In bandit problems as well as in Monte-Carlo tree search (MCTS), variance-based policies such as UCB-V are reported to show better performance in practice than policies that ignore variance information, such as UCB1. For bandits, UCB-V was proved to exhibit somewhat better convergence properties than UCB1. In contrast, for MCTS so far no convergence guarantees have been established for UCB-V. Our first contribution is to show that UCB-V provides the same convergence guarantees in MCTS that are known for UCB1.

Another open problem with variance-based policies in MCTS is that they can only be used in conjunction with Monte-Carlo backups but not with the recently suggested and increasingly popular dynamic programming (DP) backups. This is because standard DP backups do not propagate variance information. Our second contribution is to derive update equations for the variance in DP backups, which significantly extends the applicability of variance-based policies in MCTS.

Finally, we provide an empirical analysis of UCB-V and UCB1 in two prototypical environments showing that UCB-V significantly outperforms UCB1 both with Monte-Carlo as well as with dynamic programming backups.

Introduction

Monte-Carlo tree search (MCTS) has become a standard planning method and has been successfully applied in various domains, ranging from computer Go to large-scale POMDPs (Silver et al. 2016; Browne et al. 2012). Some of the most appealing properties of MCTS are that it is easy to implement, does not require a full probabilistic model of the environment but only the ability to simulate state transitions, is suited for large-scale environments, and provides theoretical convergence guarantees.

The core idea in MCTS is to treat a sequential decision problem as a series of bandit problems (Berry and Fristedt 1985). The main difference, however, is that in bandit problems the return distributions are assumed to be stationary whereas in MCTS they are not because the return distributions vary with the tree-policy. This means that convergence properties do not necessarily carry over from the bandit setting to MCTS.

The most popular MCTS algorithm is UCT (Kocsis and Szepesvári 2006), which uses UCB1 (Auer, Cesa-Bianchi, and Fischer 2002) as tree-policy. UCB1 has proven bounds for the expected regret in the bandit setting as well as polynomial convergence guarantees for the failure probability in the MCTS setting. More recently, Audibert, Munos, and Szepesvári (2009) suggested UCB-V, which takes the empirical variance of the returns into account, and proved bounds for the expected regret in the bandit setting. In the case of MCTS, however, no convergence guarantees have been proved so far. Our first contribution in this paper is to show that UCB-V, just like UCB1, provides polynomial convergence guarantees in the MCTS setting.

Apart from the tree-policy, an important aspect of an MCTS algorithms is the employed backup method. The most common variants are Monte-Carlo (MC) backups and the more recently suggested dynamic programming (DP) backups (Keller and Helmert 2013). DP backups have become increasingly popular because they show good convergence properties in practice (see Feldman and Domshlak 2014a for a comparison). The use of variance-based policies, however, has so far been restricted to MC backups since here the variance information is readily available. In contrast, DP backups do not generally propagate variance information. Our second contribution is the derivation of update equations for the variance that enable the use of variance-based policies in conjunction with DP backups.

Finally, we evaluate UCB-V and UCB1 in different environments showing that, depending on the problem characteristics, UCB-V significantly outperforms UCB1 both with MC as well as with DP backups.

In the remainder we will discuss related work on MCTS and reinforcement learning, present the proof for the convergence guarantees of UCB-V, derive the update equations for the variance with DP backups, and present our empirical results.

Background & Related Work

Monte-Carlo Tree Search

There exists a wide variety of MCTS algorithms that differ in a number of aspects. Most of them follow a generic scheme that we reproduce in Alg. 1 for convenience. Note that some recent suggestions deviate slightly from this scheme (Keller and Helmert 2013; Feldman and Domshlak 2014b). In Alg. 1 we highlighted open parameters that need to be defined in order to produce a specific MCTS im-

Algorithm 1 MCTS: Generic algorithm with open parameters for finit-horizon non-discounted environments. Notation: $(\ )$ is a tuple; $\langle\ \rangle$ is a list, $+$ appends an element to the list, $|\ l\ |$ is the length of list l, and l_i is its i^{th} element.

Input: v_0 → *root node*
s_0 → *current state*
M → *environment model*
Output: a^* → *optimal action from root node / current state*

```
1: function MCTS(v_0, s_0, M)
2:     while time permits do
3:         (ρ, s) ← FOLLOWTREEPOLICY(v_0, s_0)
4:         R ← FOLLOWDEFAULTPOLICY(s)
5:         UPDATE(ρ, R)
6:     end while
7:     return BESTACTION(v_0)                    → open parameter
8: end function

9: function FOLLOWTREEPOLICY(v, s)
10:    ρ ← ⟨⟩
11:    do
12:        a ← TREEPOLICY(v)                     → open parameter
13:        (s', r) ← M(a, s)
14:        ρ ← ρ + ⟨(v, s, a, s', r)⟩
15:        v ← FINDNODE(v, a, s')                → open parameter
16:        s ← s'
17:    while v is not a leaf node
18:    return (ρ, s)
19: end function

20: function FOLLOWDEFAULTPOLICY(s)
21:    R ← 0
22:    repeat
23:        a ← DEFAULTPOLICY(s)                  → open parameter
24:        (s', r) ← M(a, s)
25:        R ← R + r
26:        s ← s'
27:    until s is terminal state
28:    return R
29: end function

30: function UPDATE(ρ, R)
31:    for i in |ρ|, ..., 1 do
32:        (v, s, a, s', r) ← ρ_i
33:        BACKUP(v, s, a, s', r, R)             → open parameter
34:        R ← r + R
35:    end for
36: end function
```

plementation. Two of these parameters, the TREEPOLICY and the BACKUP method, will be discussed in more detail below.

BESTACTION(v_0) selects the action that is eventually recommended – usually the action with maximum empirical mean return (see e.g. Browne et al. 2012 for alternatives).

FINDNODE(v, s, a, s') selects a child node or creates a new leaf node if the child does not exist. This procedure usually builds a tree but it can also construct directed acyclic graphs (see e.g. Saffidine, Cazenave, and Méhat 2012).

DEFAULTPOLICY(s) is a heuristic policy for initializing the return for new leaf nodes – usually the uniform policy.

TREEPOLICY(v) The tree-policy selects actions in internal nodes and has to deal with the exploration-exploitation dilemma: It has to focus on high-return branches (exploitation) but it also has to sample sub-optimal branches to some extend (exploration) to make sure the estimated returns converge to the true ones. A common choice for the tree-policy is UCB1 (Auer, Cesa-Bianchi, and Fischer 2002), which chooses actions as[1]

$$a^* = \operatorname{argmax}_a B_{(s,a)} \qquad \text{with} \tag{1}$$

$$B_{(s,a)} = \widehat{R}_{(s,a)} + 2\,C_p\sqrt{\frac{2\,\log n_s}{n_{(s,a)}}} \tag{2}$$

where $\widehat{R}_{(s,a)}$ is the mean return of action a in state s, n_s is the number of visits to state s, $n_{(s,a)}$ is the number of times action a was taken in state s, the returns are assumed to be in $[\,0,1\,]$, and the constant $C_p > 0$ controls exploration. For UCB1 Kocsis and Szepesvári (2006) proved that the probability of choosing a sub-optimal action at the root node converges to zero at a polynomial rate as the number of trials grows to infinity.

More recently, Audibert, Munos, and Szepesvári (2009) suggested UCB-V that selects actions as

$$a^* = \operatorname{argmax}_a B_{(s,a)} \qquad \text{with} \tag{3}$$

$$B_{(s,a)} = \widehat{R}_{(s,a)} + \sqrt{\frac{2\,\widetilde{R}_{(s,a)}\zeta\log n_s}{n_{(s,a)}}} + 3cb\frac{\zeta\log n_s}{n_{(s,a)}} \tag{4}$$

where $\widehat{R}_{(s,a)}$, n_s, $n_{(s,a)}$ as above, $\widetilde{R}_{(s,a)}$ is the empirical variance of the return, rewards are assumed to be in $[\,0,b\,]$, and the constants $c,\zeta > 0$ control the algorithm's behavior. For the bandit setting Audibert, Munos, and Szepesvári (2009) proved regret bounds but for the MCTS setting we are not aware of any proof similar to the one for UCB1. In Section *Bounds and Convergence Guarantees* we will adapt the proof of Kocsis and Szepesvári (2006) to show that UCB-V provides the same convergence guarantees as UCB1 in the MCTS setting.

BACKUP(v, s, a, s', r, R) The BACKUP procedure is responsible for updating node v given the transition $(s,a) \to (s',r)$ and the return R of the corresponding trial. It has to maintain the data needed for evaluating the tree-policy. In the simplest case of MC backups the BACKUP procedure maintains visit counts n_s, action counts $n_{(s,a)}$, and an estimate of the expected return $\widehat{R}_{(s,a)}$ by accumulating the average of R. In the more recently suggested DP backups (Keller

[1] We use states and actions as subscripts to remain consistent with the MCTS setting.

and Helmert 2013) the BACKUP procedure also maintains a transition model and an estimate of the expected immediate reward that are then used to calculate $\widehat{R}_{(s,a)}$ while the return samples R are ignored. MC and DP backups have significantly different characteristics that are subject of ongoing research (Feldman and Domshlak 2014a). Recently, temporal difference learning and function approximation have also been proposed as backup methods (Silver, Sutton, and Müller 2012; Guez et al. 2014). It has also been suggested to use different backup methods depending on the empirical variance of returns (Bnaya et al. 2015).

When attempting to use variance information in MCTS a major problem arises because the variance of the return is usually not maintained by the BACKUP procedure. As we discuss in Section *Variance Backups*, for MC backups the extension is straightforward whereas for DP backups this is not the case. The combination of variance-based tree-policies with DP backups has therefore not been possible so far. In this paper we close this gap by deriving general update equations for the variance with DP backups.

In conclusion, while the UCB-V policy has been established for bandits, no convergence proof for its use in MCTS exists to date. Furthermore, DP backups have to date not been extended to include variance updates thus limiting the applicability of UCB-V and other variance-based methods in MCTS.

Reinforcement Learning

The exploration-exploitation dilemma exists not only for bandits and MCTS but generally in reinforcement learning. Bayesian reinforcement learning (Vlassis et al. 2012) offers a general solution that, however, is intractable for most practical problems. Various approaches, such as R-MAX (Brafman and Tennenholtz 2003) and the Bayesian Exploration Bonus (Kolter and Ng 2009) offer near-optimal approximations most of which follow the *optimism in the face of uncertainty* principle. In this context, the variance of the expected return can be used as a measure of uncertainty, which is for instance done in *Bayesian Q-learning* (Dearden, Friedman, and Russell 1998) where both the expected return as well as its variance are estimated by performing online updates under the assumption of normally distributed returns. The variance information is then used to guide exploration either by sampling values from the corresponding distribution or based on the *value of information* of an action.

Many general ideas, such as *optimism in the face of uncertainty* or variance-based exploration, carry over from reinforcement learning to the MCTS setting. However, as opposed to reinforcement learning, in MCTS we can explore "for free" during the planning phase. It is thus important to (a) enable the use of these concepts in MCTS, which we do by deriving update equations for the variance and (b) establish convergence guarantees, which we do for the case of UCB-V.

UCB-V for Monte-Carlo Tree Search

Bounds and Convergence Guarantees

We will now extend the guarantees for UCB-V as proved by Audibert, Munos, and Szepesvári (2009) from the bandit setting to MCTS. In doing so we will closely follow the proof by Kocsis and Szepesvári (2006) for the UCB1 policy showing that both policies exhibit the same convergence guarantees in the MCTS setting.

Let there be K arms with return $R_{k,i}$ in the i^{th} play whose estimated expected return and estimated variance of the return after n plays are

$$\widehat{R}_{k,n}= \tfrac{1}{n}\sum_{i=1}^{n} R_{k,i}\ ,\ \widetilde{R}_{k,n}= \tfrac{1}{n}\sum_{i=1}^{n}\left(R_{k,i}-\widehat{R}_{k,n}\right)^2 .$$

We assume that $R_{k,i}$ are independently and identically distributed and the expected values of $\widehat{R}_{k,n}$ and $\widetilde{R}_{k,n}$ converge

$$\begin{aligned}
&\mu_{k,n}= \mathbb{E}[\![\widehat{R}_{k,n}]\!]\ , && \sigma^2_{k,n}= \mathbb{E}[\![(\mu_{k,n}-\widehat{R}_{k,n})^2]\!]\ ,\\
&\mu_k\ = \lim_{n\to\infty}\mu_{k,n}\ ,\ \sigma^2_k\ = \lim_{n\to\infty}\sigma^2_{k,n}\ ,\\
&\delta_{k,n}= \mu_{k,n}-\mu_k\ .
\end{aligned}$$

We denote quantities associated with the optimal arm with an asterisk and define $\Delta_k = \mu^* - \mu_k$. The action selection rule of UCB-V is[2]

$$I_n = \operatorname*{argmax}_{k\in\{1,\dots,K\}} B_{k,n_k,n} \qquad \text{with}$$

$$B_{k,n_k,n} = \widehat{R}_{k,n_k} + \sqrt{\frac{2\widetilde{R}_{k,n_k}\zeta\log(n)}{n_k}} + 3bc\frac{\zeta\log(n)}{n_k}$$

where n is the total number of plays, n_k is the number of plays for the k^{th} arm, $b = R_{\max}$, and c, ζ are exploration parameters. Similar to Kocsis and Szepesvári (2006) we will assume that the error of the expected values of $\widehat{R}_{k,n}$ and $\widetilde{R}_{k,n}$ can be bounded and use this assumption for all results in the paper without explicitly repeating it:

Assumption 1. *For any $\epsilon > 0$, and $\tau \geq 1$, there exists $N_0(\epsilon,\tau)$ such that for all $n \geq N_0(\epsilon,\tau)$: $|\delta_{k,n}| \leq \epsilon\,\Delta_k/2$, $|\delta^*_n| \leq \epsilon\,\Delta_k/2$, and $\sigma^2_{k,n} \leq \tau\,\sigma^2_k$.*

We begin by repeating Theorem 1 in (Audibert, Munos, and Szepesvári 2009), which we use in what follows.

Theorem 1. *For any $t \in \mathbb{N}$ and $x > 0$*

$$P\left(|\widehat{R}_{k,t}-\mu| \geq \sqrt{\frac{2\widetilde{R}_{k,t}x}{t}} + 3b\frac{x}{t}\right) \leq 3e^{-x}\ . \qquad (5)$$

On the other hand,

$$P\left(|\widehat{R}_{k,s}-\mu| \geq \sqrt{\frac{2\widetilde{R}_{k,s}x}{s}} + 3b\frac{x}{s}\right) \leq \beta(x,t) \qquad (6)$$

holds for all $s \in \{1,\dots,t\}$ with

$$\beta(x,t) = 3\inf_{1<\alpha\leq 3}\left(\frac{\log t}{\log\alpha}\wedge t\right)e^{-x/\alpha}\ ,$$

where $u \wedge v$ denotes the minimum of u and v.

[2] We switch back from the notation used in Eq. (4) to a notation that ignores the state s and instead includes the number of samples.

The following first result extends Lemma 1 in (Audibert, Munos, and Szepesvári 2009).

Lemma 1. *Let*

$$u = \left\lceil 8(c \vee 1)\Big(\frac{\sigma_k^2}{\tau\Delta_k^2} + \frac{2b}{\tau\Delta_k}\Big)\zeta_n \right\rceil ,$$

where $u \vee v$ denotes the maximum of u and v. Then, for $u \leq n_k \leq t \leq n$,

$$P(B_{k,n_k,t} > \mu_t^*) \leq 2e^{-\frac{n_k\tau\Delta_k^2}{8\sigma_k^2 + 4b\Delta_k/3}} .$$

Proof. In Appendix II. □

We define $A(n,\epsilon,\tau) = N_0(\epsilon,\tau) \vee u$ and bound the number of plays of an arm k for non-stationary multi-armed bandits:

Theorem 2. *Applying UCB-V for non-stationary multi-armed bandits, we can bound the expected number of plays of arm k as*

$$\mathbb{E}[T_k(n)] \leq A(n,\epsilon,\tau) + ne^{-(c\vee 1)\zeta_n}\Big(\frac{24\sigma_k^2}{\tau\Delta_k^2} + \frac{4b}{\tau\Delta_k}\Big) + \ldots$$
$$\ldots + \sum_{t=u+1}^{n} \beta\big((c\wedge 1)\zeta_t, t\big) \tag{7}$$

Proof. In Appendix II. □

The following theorem is the counter-part of Theorem 2 in UCT (Kocsis and Szepesvári 2006). This bound is different in that it takes the variance of the return into account.

Theorem 3. *The expected regret is bounded by*

$$\big|\mathbb{E}[\widehat{R}_n] - \mu^*\big| \leq |\delta_n^*| + O\Bigg(\frac{N_0(\epsilon,\tau)}{n} + \ldots$$
$$\ldots + \frac{\sum_k \Big[\frac{\tau\sigma_k^2}{(1-2\epsilon)\Delta_k} + \frac{\sigma_k^2}{b^2\Delta_k} + 2b + 2\Big]\log(n)}{n}\Bigg) \tag{8}$$

Proof. The proof follows the same simple derivation of Theorem 2 in UCT (Kocsis and Szepesvári 2006), then follows the same trick to bound the sum appearing in Theorem 2. □

Theorem 4. *Under the assumptions of Lemma 1 and Theorems 2 and 3, the failure probability converges to zero*

$$\lim_{n\to\infty} P(I_n \neq k^*) = 0$$

Proof. The proof follows exactly the proof of Theorem 5 in (Kocsis and Szepesvári 2006). □

We are now in the position to prove the most important theoretical result of UCB-V applied to MCTS. Although the result in Theorem 3 takes into account the variance of the reward distribution, we prefer to upper-bound the expected regret by a different term for simplicity. As the sum contains only constants and runs over $k \in \{1,\ldots,K\}$ we can upper-bound it as

$$\big|\mathbb{E}[\widehat{R}_n] - \mu^*\big| \leq |\delta_n^*| + O\Bigg(\frac{K\log(n) + N_0(\epsilon,\tau)}{n}\Bigg) , \tag{9}$$

which leads to the final result.

Theorem 5. *Applying UCB-V as tree-policy in a search tree of depth D, with branching factor K, and returns in $[0,b]$ the expected regret at the root node is bounded by*

$$O\Bigg(\frac{KD\log(n) + K^D}{n}\Bigg) .$$

At the same time, the probability of choosing a sub-optimal action at root node converges to zero in polynomial time.

Proof. Using the simplified bound in Eq. (9), the proof follows similarly to the proof of Theorem 6 in (Kocsis and Szepesvári 2006). □

Variance Backups

The BACKUP(v,s,a,s',r,R)-routine has to update node v's data based on the transition information $(s,a) \to (s',r)$ and the return R of the corresponding trial. In order to use variance-based tree-policies we need to use this information to not only maintain an estimate of the expected return but also of its variance.

For MC backups this extension is trivial since it suffices to maintain quadratic statistics of the return and estimate the variance as

$$\widetilde{R}_{(s,a)} = \mathbb{E}[\![R_{(s,a)}^2]\!] - \mathbb{E}[\![R_{(s,a)}]\!] . \tag{10}$$

For DP backups on the other hand we need to propagate the variance up the tree just as we do for the expected value. The value of action a in state s is defined as the expected return

$$Q_{(s,a)} = \mathbb{E}[\![R_{(s,a)}]\!] , \tag{11}$$

so that when we estimate Q from the available samples it is itself a random variable whose expected value and variance we denote by $\widehat{Q}$ and $\widetilde{Q}$, respectively. The DP updates for the value are defined as

$$Q_{(s,a)} = \sum_{s'} p_{(s'|s,a)}\big(r_{(s,a,s')} + \gamma\, V_{s'}\big) \tag{12}$$

$$\text{and} \qquad V_s = \sum_a \pi_{(a|s)}\, Q_{(s,a)} \tag{13}$$

where V_s is the state value; $\pi_{(a|s)}$ is the probability of choosing action a in state s; $p_{(s'|s,a)}$ is the probability of transitioning from state s to state s' upon taking action a; $r_{(s,a,s')}$ is the expected reward for such a transition; and $0 \leq \gamma \leq 1$ is the discount factor. p and r are random variables whose mean and variance can be estimated from data while π is chosen by the algorithm. Eqs. (12) and (13) carry over to the expected values by simply replacing all variables with their estimated values, which gives the standard DP backups used in MCTS. The implicit assumption here is that variables associated to different states are independent, which we will also assume from now on. In order to estimate the variance we have to use Eqs. (12) and (13) and explicitly write out the expectations

$$\widetilde{Q}_{(s,a)} = \mathbb{E}[\![Q^2_{(s,a)}]\!] - \mathbb{E}[\![Q_{(s,a)}]\!]^2 \tag{14}$$

$$= \sum_{s'} \left[\widehat{p}^2_{(s'|s,a)} + \widetilde{p}_{(s'|s,a)}\right]\left[\widetilde{r}_{(s,a,s')} + \gamma^2\,\widetilde{V}_{s'}\right] + \dots$$

$$\dots + \sum_{s',s''} \widetilde{p}_{(s'/s''|s,a)}\left[\widehat{r}_{(s,a,s')} + \gamma\,\widehat{V}_{s'}\right]\left[\widehat{r}_{(s,a,s'')} + \gamma\,\widehat{V}_{s''}\right] \tag{15}$$

$$\widetilde{V}_s = \sum_a \pi^2_{(a|s)}\,\widetilde{Q}_{(s,a)} \tag{16}$$

where $\widetilde{p}$ and $\widetilde{r}$ are the (co)variances of p and r. We defer the full derivation to the Appendix I. For the immediate reward, r, we maintain linear and quadratic statistics to compute its mean and variance. For the transition probabilities, p, we maintain transition counts, from which the expected value $\widehat{p}$ and variance $\widetilde{p}$ can be computed, assuming p to be Dirichlet distributed.

Experiments

We performed experiments in various domains combining UCB-V and UCB1 with MC and DP backups. Our evaluations revealed that, depending on the problem characteristics, each of the four possibilities may significantly outperform the others. While an exhaustive presentation and discussion of all results is beyond the scope of this paper, we present two exemplary cases where UCB-V with MC and DP backups, respectively, outperforms the alternatives and discuss possible explanations. Fig. 1 shows for both cases the probability of choosing the optimal action at the root node as a function of the number of rollouts. The optimal action a^* is known for each problem and its probability is computed as relative frequency of a^* actually being recommended by the planner (i.e. having the maximum empirical mean return) after each given run. In all experiments we used $c = 1$ and $\zeta = 1.2$ for UCB-V and $C_p = 1/\sqrt{2}$ for UCB1, for which regret bounds were proved in (Audibert, Munos, and Szepesvári 2009) and (Auer, Cesa-Bianchi, and Fischer 2002), respectively.

Stochastic1D In this environment the agent moves along a line and receives a terminal reward after a fixed time T. Each action moves the agent $\{-k, \dots, k\}$ steps along the line, so there are $2\,k+1$ actions to choose from and after T steps the agent may be at any position $x \in \{-k\,T, \dots, k\,T\}$. When performing an action, with probability α the agent actually performs the chosen action and with probability $1-\alpha$ it performs a random action. After T time steps, with probability β the agent receives a terminal reward and with probability $1-\beta$ it receives a reward of zero instead. The terminal rewards lie in $[0, 1]$ and scale linearly with the terminal position x

$$r = \frac{x + k\,T}{2\,T\,k}\ . \tag{17}$$

The optimal policy thus is to always choose action k. Results in Fig. 1(a) are averaged over 10000 runs for parameters $k = 3$, $T = 10$, $\alpha = 0.6$, $\beta = 0.5$.

Two properties of *Stochastic1D* make UCB-V in conjunction with MC backups favorable. First, in this environment MC backups with a uniform rollout policy will in expectation yield the optimal policy. This allows to take advantage of the more robust convergence properties of MC backups as compared to DP backups. Second, the optimal reward also has the highest variance. Since UCB-V is biased towards high-variance branches this favors UCB-V over UCB1.[3]

NastyStochastic1D This environment is identical to the *Stochastic1D* environment except for the magnitude of the terminal rewards, which are "misleading" in this case. The maximum reward of 1 is still received when the agent ends at position $k\,T$, however, the second-best reward is received at position $-k\,T$ and then decreases linearly until reaching the minimum of 0 when the agent misses the optimal reward by one step and ends at position $k\,T-1$

$$r = \begin{cases} 1 & \text{if } x = k\,T \\ \frac{k\,T-x-1}{2\,T\,k} & \text{else .} \end{cases} \tag{18}$$

The optimal policy is the same as in the *Stochastic1D* environment. Results in Fig. 1(b) are averaged over 4000 runs for parameters $k = 1$, $T = 3$, $\alpha = 0.9$, $\beta = 1$.

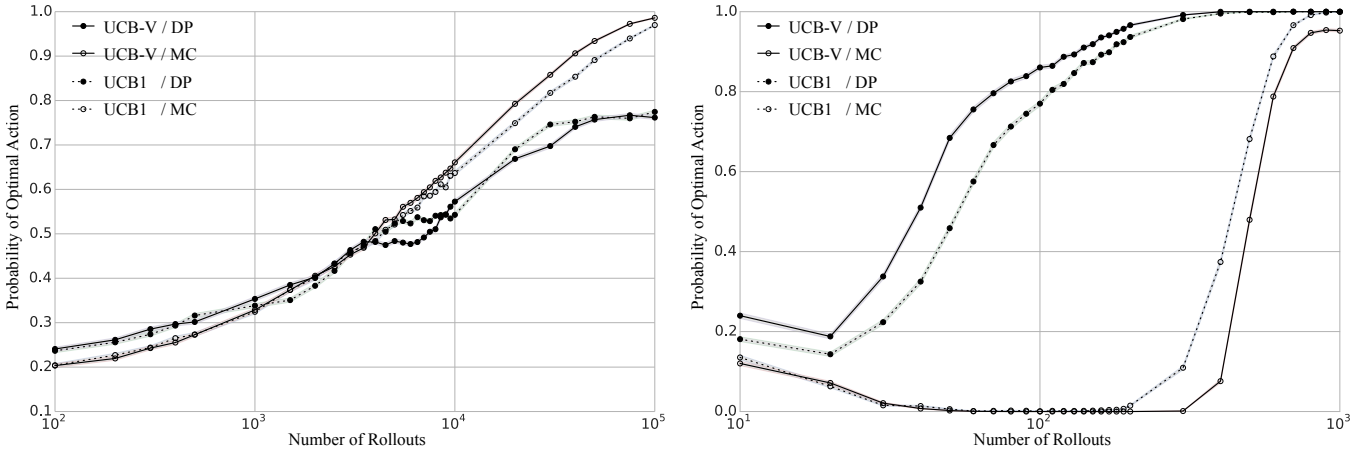
In *NastyStochastic1D*, again, the maximum reward also has the maximum variance, favoring UCB-V over UCB1. This time, however, MC backups with a uniform rollout policy will in expectation result in a sub-optimal policy that guides the agent away from the optimal path – note the negative slope in the initial planning phase in Fig. 1(b). In this situation, the ability of DP backups to quickly "switch" to a different path gives them a clear advantage over MC backups.

The presented results are examples showing that the best choice (in this case UCB-V versus UCB1 and MC versus DP) strongly depends on the characteristics of the problem at hand. It is therefore important to be able to freely choose the method that suits the problem best and to be assured of convergence guarantees. In this respect, our paper makes an important contribution for the case of variance-based tree-policy in general, and UCB-V in particular.

Conclusion

We showed that the variance-based policy UCB-V (Audibert, Munos, and Szepesvári 2009) provides the same theoretical guarantees for Monte-Carlo tree search as the widely used UCB1 policy, namely, a bounded expected regret and polynomial convergence of the failure probability. We additionally derived update equations for the variance allowing to combine variance-based tree-policies with dynamic programming backups, which was not possible so far. In our experimental evaluations we demonstrate that, depending on the problem characteristics, UCB-V significantly outperforms UCB1.

[3] Giving high rewards a low variance and vice versa will in general deteriorate the performance of UCB-V as compared to UCB1.



(a) Stochastic1D (10000 runs) (b) NastyStochastic1D (4000 runs)

Figure 1: Experiments. The plots show the probability of choosing the optimal action at the root node as a function of the number of rollouts. Solid lines correspond to UCB-V policy, dashed lines to UCB1. Filled circles correspond to dynamic programming backups, open circles to Monte-Carlo backups. Note that due to the large number of runs the error bands are barely visible.

Appendix I: Derivation of Variance Updates

We use the following notation

$\widehat{x} = \mathbb{E}\big[\!\big[\, x \,\big]\!\big]_x$	expected value of x
$\widetilde{x} = \mathbb{E}\big[\!\big[\, x^2 \,\big]\!\big]_x - \widehat{x}^{\,2}$	variance of x
$\mathrm{cov}(x,y) = \mathbb{E}\big[\!\big[xy\big]\!\big]_{x,y} - \widehat{x}\widehat{y}$	covariance of x and y .

We assume π and γ to be scalar variables (π may still represent a non-deterministic policy). V, Q, p, and r are random variables that are assumed independent so that all covariance terms vanish (i.e. only the diagonal variance terms remain). The only exception to this are the transition probabilities for the same state-action pair but with a different target state, where we use

$$\mathrm{cov}(p_{(s'|s,a)}, p_{(s''|s,a)}) = \widetilde{p}_{(s'/s''|s,a)} \tag{19}$$

as a more compact notation. For the variance of the state value V we get

$$\widetilde{V}_s = \mathbb{E}\Big[\!\Big[\Big[\sum_a \pi_{(a|s)}\, Q_{(s,a)} \Big]^2 \Big]\!\Big]_{r,p} - \widehat{V}_s^{\,2} \tag{20}$$

$$= \sum_{a,a'} \pi_{(a|s)}\, \pi_{(a'|s)}\, \mathbb{E}\Big[\!\Big[\, Q_{(s,a)}\, Q_{(s,a')} \,\Big]\!\Big]_{r,p} - \widehat{V}_s^{\,2} \tag{21}$$

$$= \sum_a \pi_{(a|s)}^2\, \widetilde{Q}_{(s,a)} \ . \tag{16}$$

The variance of the state-action value is

$$\widetilde{Q}_{(s,a)} = \tag{22}$$

$$= \mathbb{E}\Big[\!\Big[\Big[\sum_{s'} p_{(s'|s,a)} \big(r_{(s,a,s')} + \gamma\, V_{s'} \big) \Big]^2 \Big]\!\Big]_{r,p} - \widehat{Q}_{(s,a)}^2 \tag{23}$$

$$= \sum_{s',s''} \mathbb{E}\big[\!\big[p_{(s'|s,a)}\, p_{(s''|s,a)} \big]\!\big]_{r,p}\, \mathbb{E}\big[\!\big[\, \gamma^2\, V_{s'}\, V_{s''} + r_{(s,a,s')}\, r_{(s,a,s'')} + \gamma\, r_{(s,a,s')}\, V_{s''} + \gamma\, r_{(s,a,s'')}\, V_{s'} \,\big]\!\big]_{r,p} - \widehat{Q}_{(s,a)}^2 \tag{24}$$

$$= \sum_{s',s''} \Big[\widehat{p}_{(s'|s,a)}\, \widehat{p}_{(s''|s,a)} + \mathrm{cov}(p_{(s'|s,a)}, p_{(s''|s,a)}) \Big] \Big[\gamma^2 \big[\, \widehat{V}_{s'}\, \widehat{V}_{s''} + \mathrm{cov}(V_{s'}, V_{s''}) \,\big] + \dots \tag{25a}$$

$$\dots + \big[\, \widehat{r}_{(s,a,s')}\, \widehat{r}_{(s,a,s'')} + \mathrm{cov}(r_{(s,a,s')}, r_{(s,a,s'')}) \,\big] + \gamma\, \widehat{r}_{(s,a,s')}\, \widehat{V}_{s''} + \gamma\, \widehat{r}_{(s,a,s'')}\, \widehat{V}_{s'} \Big] - \widehat{Q}_{(s,a)}^2 \tag{25b}$$

$$= \sum_{s',s''} \mathrm{cov}(p_{(s'|s,a)}, p_{(s''|s,a)}) \Big[\gamma^2\, \widehat{V}_{s'}\, \widehat{V}_{s''} + \widehat{r}_{(s,a,s')}\, \widehat{r}_{(s,a,s'')} + \gamma\, \widehat{r}_{(s,a,s')}\, \widehat{V}_{s''} + \gamma\, \widehat{r}_{(s,a,s'')}\, \widehat{V}_{s'} \Big] + \dots \tag{26a}$$

$$\dots + \sum_{s',s''} \Big[\widehat{p}_{(s'|s,a)}\, \widehat{p}_{(s''|s,a)} + \mathrm{cov}(p_{(s'|s,a)}, p_{(s''|s,a)}) \Big] \Big[\gamma^2\, \mathrm{cov}(V_{s'}, V_{s''}) + \mathrm{cov}(r_{(s,a,s')}, r_{(s,a,s'')}) \Big] + \dots \tag{26b}$$

$$\dots + \underbrace{\sum_{s',s''} \widehat{p}_{(s'|s,a)}\, \widehat{p}_{(s''|s,a)} \Big[\gamma^2\, \widehat{V}_{s'}\, \widehat{V}_{s''} + \widehat{r}_{(s,a,s')}\, \widehat{r}_{(s,a,s'')} + \gamma\, \widehat{r}_{(s,a,s')}\, \widehat{V}_{s''} + \gamma\, \widehat{r}_{(s,a,s'')}\, \widehat{V}_{s'} \Big]}_{=0} - \widehat{Q}_{(s,a)}^2 \tag{26c}$$

$$= \sum_{s'} \Big[\widehat{p}^{\,2}_{(s'|s,a)} + \widetilde{p}_{(s'|s,a)}\Big]\Big[\widetilde{r}_{(s,a,s')} + \gamma^2\,\widetilde{V}_{s'}\Big] + \sum_{s',s''} \widetilde{p}_{(s'/s''|s,a)}\Big[\widehat{r}_{(s,a,s')} + \gamma\,\widehat{V}_{s'}\Big]\Big[\widehat{r}_{(s,a,s'')} + \gamma\,\widehat{V}_{s''}\Big] \tag{15}$$

where in lines 26a–26c we arrange terms such that in 26b the terms with $s \neq s'$ vanish because the covariances of r and V then vanish by assumption, and in 26c the first part exactly reproduces $\widehat{Q}^{\,2}_{(s,a)}$ so that the complete line cancels out. Using the simplified notation given in Eq. (19) for the covariance of p in 26a we finally reproduce Eq. (15).

Appendix II: Proofs for Lemma 1 and Theorem 2

Proof. (Lemma 1) We define $\zeta_n = \zeta \log(n)$. From the definition of $B_{k,n_k,t}$, we have

$$\begin{aligned}
&P(B_{k,n_k,t} > \mu_t^*) \\
&= P\Big(\widehat{R}_{k,n_k} + \sqrt{\frac{2\widetilde{R}_{k,n_k}\zeta_t}{n_k}} + 3bc\frac{\zeta_t}{n_k} > \mu^* + \delta_t^*\Big) \\
&= P\Big(\widehat{R}_{k,n_k} + \sqrt{\frac{2\widetilde{R}_{k,n_k}\zeta_t}{n_k}} + 3bc\frac{\zeta_t}{n_k} > \mu_k + \Delta_k + \delta_t^*\Big) \\
&= P\Big(\widehat{R}_{k,n_k} + \sqrt{\frac{2\widetilde{R}_{k,n_k}\zeta_t}{n_k}} + 3bc\frac{\zeta_t}{n_k} > \mu_{k,t} + \delta_{k,t} + \Delta_k + \delta_t^*\Big) \\
&\leq P\Big(\widehat{R}_{k,n_k} + \sqrt{\frac{2\widetilde{R}_{k,n_k}\zeta_t}{n_k}} + 3bc\frac{\zeta_t}{n_k} > \mu_{k,t} + \Delta_k - \epsilon\Delta_k\Big) \\
&\Big(\text{using the fact that } |\delta_{k,t}| \leq \epsilon\Delta_k/2 \text{ and } |\delta_t^*| \leq \epsilon\Delta_k/2 \text{ for } l \geq N_0(\epsilon,\tau)\Big) \\
&\leq P\Big(\widehat{R}_{k,n_k} + \sqrt{\frac{2[\sigma_{k,t}^2 + b\tau\Delta_k/2]\zeta_t}{n_k}} + 3bc\frac{\zeta_t}{n_k} > \mu_{k,t} + (1-\epsilon)\Delta_k\Big) \\
&+ P(\widetilde{R}_{k,n_k,t} \geq \sigma_{k,t}^2 + b\tau\Delta_k/2)\,.
\end{aligned}$$

For the second term,

$$\begin{aligned}
\widetilde{R}_{k,n_k,n} &= \frac{1}{n_k}\sum_{j=1}^{n_k}(R_{k,j} - \mu_{k,t})^2 - (\mu_{k,t} - \widehat{R}_{k,n_k,t})^2 \\
&\leq \frac{1}{n_k}\sum_{j=1}^{n_k}(R_{k,j} - \mu_{k,t})^2\,,
\end{aligned}$$

hence,

$$P(\widetilde{R}_{k,n_k,t} \geq \sigma_{k,t}^2 + b\tau\Delta_k/2) \leq P\left(\frac{\sum_{j=1}^{n_k}(R_{k,j} - \mu_{k,t})^2}{n_k} - \sigma_{k,t}^2 \geq \frac{b\tau\Delta_k}{2}\right).$$

For the first term, we use the fact that $u \leq n_k \leq t \leq n$, $\sigma_{k,t}^2 \leq \tau\sigma_k^2$, and the definition of u to derive

$$\begin{aligned}
&\sqrt{\frac{2[\sigma_{k,t}^2 + b\tau\Delta_k/2]\zeta_t}{n_k}} + 3bc\frac{\zeta_t}{n_k} \leq \sqrt{\frac{[2\tau\sigma_k^2 + b\tau\Delta_k]\zeta_n}{u}} + 3bc\frac{\zeta_n}{u} \\
&\leq \sqrt{\frac{[2\tau\sigma_k^2 + b\tau\Delta_k]\tau\Delta_k^2}{8(\sigma_k^2 + 2b\Delta_k)}} + 3b\frac{\tau\Delta_k^2}{8(\sigma_k^2 + 2b\Delta_k)} \\
&= \frac{\tau\Delta_k}{2}\left(\sqrt{\frac{[2\sigma_k^2 + b\Delta_k]}{(2\sigma_k^2 + 4b\Delta_k)}} + 3b\frac{\Delta_k}{(4\sigma_k^2 + 8b\Delta_k)}\right)
\end{aligned}$$

$$= \frac{\tau\Delta_k}{2}\left(1 - \frac{1}{2}\left[1 - \sqrt{\frac{[2\sigma_k^2 + b\Delta_k]}{(2\sigma_k^2 + 4b\Delta_k)}}\right]^2\right)$$
$$\leq \frac{\tau\Delta_k}{2} .$$

Hence,

$$P\Big(\widehat{R}_{k,n_k} + \sqrt{\frac{2[\sigma_{k,t}^2 + b\tau\Delta_k/2]\zeta_t}{n_k}} + 3bc\frac{\zeta_t}{n_k} > \mu_{k,t} + (1-\epsilon)\Delta_k\Big)$$
$$\leq P\Big(B_{k,n_k,t} - \mu_{k,t} > \frac{(\tau - 2\epsilon)\Delta_k}{2}\Big) .$$

Using Bernstein's inequality twice, we obtain

$$P(B_{k,n_k,t} > \mu_t^*) \leq e^{-\frac{n_k(\tau-2\epsilon)^2\Delta_k^2}{8\sigma_{k,t}^2 + 4b(\tau-2\epsilon)\Delta_k/3}} + e^{-\frac{n_k b^2\tau^2\Delta_k^2}{8\sigma_{k,t}^2 + 4b^2\tau\Delta_k/3}}$$
$$\leq e^{-\frac{n_k(\tau-2\epsilon)^2\Delta_k^2}{8\tau\sigma_k^2 + 4b(\tau-2\epsilon)\Delta_k/3}} + e^{-\frac{n_k b^2\tau\Delta_k^2}{8\sigma_k^2 + 4b^2\Delta_k/3}} \text{ (the fact: } \sigma_{k,t}^2 \leq \tau\sigma_k^2)$$
$$\leq 2e^{-\frac{n_k\tau\Delta_k^2}{8\sigma_k^2 + 4b\Delta_k/3}} .$$

□

Proof. (Theorem 2) Similar to the proofs of Theorems 2 and 3 in (Audibert, Munos, and Szepesvári 2009), Theorem 1 in (Auer, Cesa-Bianchi, and Fischer 2002), and Theorem 1 in (Kocsis and Szepesvári 2006), the number of plays of a suboptimal arm k until time n for arbitrary u is

$$\mathbb{E}[T_k(n)] = \mathbb{E}\Big[\sum_{t=1}^{n}\mathbb{I}\{I_t = k\}\Big]$$
$$\leq u + \sum_{t=u+1}^{n}\sum_{n_k=u}^{t-1} P(B_{k,n_k,t} > \mu_t^*) + \sum_{t=u}^{n}\sum_{n_k=1}^{t-1} P(B_{k^*,n_k,t} \leq \mu_t^*) .$$

The last term is bounded using Theorem 1. The second term is bounded as in Lemma 1. Using the same simplifying trick as in the proof of Lemma 1 in (Audibert, Munos, and Szepesvári 2009), we obtain the final result as

$$\mathbb{E}[T_k(n)] \leq u + \sum_{t=u+1}^{n}\sum_{n_k=u}^{t-1} 2e^{-\frac{n_k\tau\Delta_k^2}{8\sigma_k^2 + 4b\Delta_k/3}} + \sum_{t=u+1}^{n}\beta\big((c\wedge 1)\zeta_t, t\big)$$
$$\leq u + \sum_{t=u+1}^{n} 2\frac{e^{-\frac{u\tau\Delta_k^2}{8\sigma_k^2 + 4b\Delta_k/3}}}{1 - e^{-\frac{\tau\Delta_k^2}{8\sigma_k^2 + 4b\Delta_k/3}}} + \sum_{t=u+1}^{n}\beta\big((c\wedge 1)\zeta_t, t\big)$$
$$\leq u + \sum_{t=u+1}^{n}\Big(\frac{24\sigma_k^2}{\tau\Delta_k^2} + \frac{4b}{\tau\Delta_k}\Big)e^{-\frac{u\tau\Delta_k^2}{8\sigma_k^2 + 4b\Delta_k/3}} + \sum_{t=u+1}^{n}\beta\big((c\wedge 1)\zeta_t, t\big)$$
(because $1 - e^{-x} \geq 2x/3$)
$$\leq A(n,\epsilon,\tau) + ne^{-(c\vee 1)\zeta_n}\Big(\frac{24\sigma_k^2}{\tau\Delta_k^2} + \frac{4b}{\tau\Delta_k}\Big) + \sum_{t=u+1}^{n}\beta\big((c\wedge 1)\zeta_t, t\big)$$

where u satisfies the condition in Lemma 1. □

References

Audibert, J.-Y.; Munos, R.; and Szepesvári, C. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410(19):1876–1902.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47:235–256.

Berry, D. A., and Fristedt, B. 1985. *Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)*. Springer.

Bnaya, Z.; Palombo, A.; Puzis, R.; and Felner, A. 2015. Confidence backup updates for aggregating mdp state values in monte-carlo tree search. In *Eighth Annual Symposium on Combinatorial Search*.

Brafman, R. I., and Tennenholtz, M. 2003. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research* 3:213–231.

Browne, C. B.; Powley, E.; Whitehouse, D.; Lucas, S. M.; Cowling, P. I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; and Colton, S. 2012. A survey of monte carlo tree search methods. *Computational Intelligence and AI in Games, IEEE Transactions on* 4(1):1–43.

Dearden, R.; Friedman, N.; and Russell, S. 1998. Bayesian q-learning. In *AAAI/IAAI*, 761–768.

Feldman, Z., and Domshlak, C. 2014a. Monte-carlo tree search: To mc or to dp? *Models and Paradigms for Planning under Uncertainty: a Broad Perspective* 11.

Feldman, Z., and Domshlak, C. 2014b. On mabs and separation of concerns in monte-carlo planning for mdps. In *Twenty-Fourth International Conference on Automated Planning and Scheduling*.

Guez, A.; Heess, N.; Silver, D.; and Dayan, P. 2014. Bayes-adaptive simulation-based search with value function approximation. In *Advances in Neural Information Processing Systems*, 451–459.

Keller, T., and Helmert, M. 2013. Trial-based heuristic tree search for finite horizon mdps. In *ICAPS*.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*. Springer. 282–293.

Kolter, J. Z., and Ng, A. Y. 2009. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 513–520. ACM.

Saffidine, A.; Cazenave, T.; and Méhat, J. 2012. Ucd: Upper confidence bound for rooted directed acyclic graphs. *Knowledge-Based Systems* 34:26–33.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Silver, D.; Sutton, R. S.; and Müller, M. 2012. Temporal-difference search in computer go. *Machine learning* 87(2):183–219.

Vlassis, N.; Ghavamzadeh, M.; Mannor, S.; and Poupart, P. 2012. Bayesian reinforcement learning. In *Reinforcement Learning*. Springer. 359–386.