Continuous Upper Confidence Trees with Polynomial Exploration - Consistency

David Auger[1], Adrien Couëtoux[2], and Olivier Teytaud[2]

[1] AlCAAP, Laboratoire PRiSM, Bât. Descartes
Université de Versailles Saint-Quentin-en-Yvelines
45 avenue des États-Unis, F-78035 Versailles Cedex, France
david.auger@prism.uvsq.fr
[2] TAO, Lri, UMR CNRS 8623, Bat. 490
Université Paris-Sud, F-91405 Orsay Cedex
adrien.couetoux@gmail.com, olivier.teytaud@gmail.com

Abstract. Upper Confidence Trees (UCT) are now a well known algorithm for sequential decision making; it is a provably consistent variant of Monte-Carlo Tree Search. However, the consistency is only proved in a the case where the action space is finite. We here propose a proof in the case of fully observable Markov Decision Processes with bounded horizon, possibly including infinitely many states, infinite action space and arbitrary stochastic transition kernels. We illustrate the consistency on two benchmark problems, one being a legacy toy problem, the other a more challenging one, the famous energy unit commitment problem.

Keywords: Upper Confidence Trees, Consistency Proof, Infinite Markov Decision Process, Unit commitment

1 State of the Art and Outline of the Paper

It is known that partially observable Markov Decision Processes are undecidable, even with finite state space (see [15]). With full observation, they become decidable; Monte-Carlo Tree Search (MCTS, [9]) is a recent well known solver for this case, with impressive results in many cases, in particular the game of Go [14]. Its most famous variant, Upper Confidence Trees [13] provides provably consistent solvers in the finite case. We here show that Upper Confidence Tree can be slightly adapted to become consistent in the more general finite horizon case, even with infinite state space and infinite action space.

Recent impressive results in the field of planning with MCTS variants in continuous MDP have been published; most of them, as far as we know, rely on a discretization of the action space. This the case of HOOT [16] and HOLOP [17] that both rely on the HOO algorithm, introduced in [5]. HOO is a bandit algorithm that deals with continuous arms by using a tree of coverings of the action space. Other notable contributions using a discretization of the action space are [12] and [1]. What these methods have in common is the assumption that the action space is continuous, but that we have enough knowledge about it to divide it in a certain number of equally spaced actions. Or, in the case of HOO, it is required to have a compact action space with known bounds. In toy benchmark problems like inverted pendulum, this is straightforward. However, in more realistic applications, this can be difficult. This is the case of the unit commitment problem, as described in [3] , where the agent needs to decide at each time step how to use a wide array of energy production facilities: water stocks, thermal plants, nuclear plants, etc. This problem has an action space that cannot be easily discretized. First, it has both discrete and continuous components (some power plants having a minimal energy output). Second, there are many operational constraints, making the action space non convex, and the bounds hard to find. In practice, finding feasible actions can come down to adding noise to the objective function of a simplified version of the problem, applying a Linear Programming method on said simplified problem, and using the result as a feasible action. There are many other options to sample a feasible action, but raw discretization is not one of them.

In this work, we investigate the consistency of a method that does not require any knowledge about the action space itself. The only assumption made is that we have access to a black box action sampler. Further details on the assumptions made are found below.

Section 2 introduces notations and specifies the setting of the Markov Decision Processes that we consider. In Section 3, we define our PUCT (Polynomial Upper Confidence Trees) algorithm. Section 4 gives the main consistency result, with convergence rate. The proof of this result is divided in three parts, which are Sections 5, 6 and 7. Section 8 presents experimental results. Section 9 concludes.

2 Specification of the Makov Decision Tree Setting

We use the classical terminology of Markov Decision Processes. In this framework, a *player* has to make sequential decisions until the process stops: he is then given a *reward*. As usual, the goal of the player is to maximize the expected reward. This paper considers the general case where the process, also called transition, is a fully observable MDP, with finite horizon, and no cycles. *In this setting, the only things available to the agent are a simulator, or transition function, and an action sampler.*

As per usual in this setting, there is a state space and an action space. To build a tree in the stochastic setting, we choose to build it with two distinct and alternated types of nodes:

- decision nodes, where a decision needs to be made, are generally noted z. The intuition is that they correspond to a certain state where the agent might be.
- random nodes, where the transition can be called, are noted $w = (z, a)$. They correspond to the case where the agent was in state z and decided to take action a (sometimes called post-decision state).

The tree will have a unique root decision node r, the initial state where the agent starts. We define the depth of a node as half the distance from this node to the root in the tree. Hence decision nodes have integer depth while random nodes have semi-integer depth, e.g. to access a node of depth 2 we have the sequence of nodes root=decision(depth 0) - random (0.5) - decision (1) - random (1.5) - decision (2). Leaves are assumed to all have the same integer depth, denoted $d_{\max}$, and bear some deterministic *reward* $r(z)$.

It is well known [2] that for each node z, there exists a value $V^*(z)$, termed optimal Bellman value, frequently used as a criterion to select the best action in sequential decision making problems. In this paper, we will use this value as a measure of optimality for actions. Given our distinction between decision nodes and random nodes, we use a natural notation for optimal Bellman values for both categories of nodes.

Let $w = (z, a)$ be a random node, and $P(z'|z, a)$ be the probability of being in node z' after taking action a in node z. Then, its optimal value is:

$$V^*(z, a) = \int_{z'} dP(z'|z, a)V^*(z') \tag{1}$$

Let z be a decision node. Then, its optimal value is defined as follows:

$$V^*(z) = \begin{cases} sup_a V^*(z, a) & \text{if } z \text{ is not a leaf,} \\ r(z) & \text{if } z \text{ is a leaf} \end{cases} \tag{2}$$

In particular, we formally define optimality of actions as follows:

Definition 1. *Let z be a non-leaf decision node, $w = (z, a)$ be a child of z, and $\epsilon > 0$. We say that the action a, i.e. the selection of node w, is optimal with precision ϵ if and only if $V(w) \geq V^*(z) - \epsilon$.*

There may be no optimal action since the number of children is infinite.

Regularity hypothesis for decision nodes This is the assumption that for any $\Delta > 0$, there is a non zero probability to sample an action that is optimal with precision Δ. More precisely, there is a $\theta > 0$ and a $p > 1$ (which remain the same during the whole simulation) such that for all $\Delta > 0$,

$$V(w = (z, a)) \geq V^*(z) - \Delta \text{ with probability at least } \min(1, \theta\Delta^p). \tag{3}$$

3 Specification of the Polynomial Upper Confidence Tree Algorithm

We refer to [13] for the detailed specification of Upper Confidence Tree; we here define our variant PUCT (Polynomial Upper Confidence Trees).

In PUCT, we sequentially repeat episodes of the MDP and use information from previous episodes in order to explore and find optimal actions in the subsequent episodes. We denote by $n(z)$, for any decision node z, the total number of times that node z has been visited after the n^{th} episode. Hence a node z has been encountered at episode n if $n(z) \geq 1$, and we always have $n = n(r)$. The notation is identical for random nodes.

We denote by $\hat{V}(z)$ the empirical average of a decision node z and $\hat{V}(z, a)$ the empirical average of a random node $w = (z, a)$. Note that if PUCT works properly, $\hat{V}(z)$ should converge to $V^*(z)$ when $n(z)$ goes to infinity.

How we select and construct children of a given node depends on two sequences of coefficients: α_d, the *progressive widening coefficient*, defined for all integer and semi-integer depths d, and e^d, the *exploration coefficient*, defined only for integer depths (i.e. decision nodes). These coefficients are defined according to Table 4. We sometimes indicate, as on Table 4, by a small "R" or "D" if a coefficient corresponds to a random or decision node, but otherwise it should be clear from the context.

```
PUCT algorithm
Input: a root node $r$, a transition function, an action sampler, a time budget, a
depth $d_{max}$, parameters $\alpha$ and $e$ for each layer
Output: an action $a$
while time budget not exhausted do
  while current node is not final do
    if current node is a decision node $z$ then
      if $\lfloor n(z)^\alpha \rfloor > \lfloor (n(z) - 1)^\alpha \rfloor$ then
        we call the action sampler and add a child $w = (z, a)$ to $z$
      else
        we choose as an action among the already visited children $(z, a)$ of $z$, the
        one that maximizes its score, defined by:

            $\hat{V}(z, a) + \sqrt{\frac{n(z)^{e(d)}}{n(z, a)}}$.        (4)

      end if
    else
      if $\lfloor n(w)^\alpha \rfloor = \lfloor (n(w) - 1)^\alpha \rfloor$ then
        we select the child of $z$ that was least visited during the simulation
      else
        we construct a new child (i.e. we call the transition function with argument $w$)
      end if
    end if
  end while
  we reached a final node $z$ with reward $r(z)$; we back propagate all the information
  in the constructed nodes, and we go back to the root node $r$.
end while
Return the most simulated child of $r$.
```

With this algorithm, we see that if a decision node z at depth d has been visited n times, then we have visited during the simulation exactly $\lfloor n^{\alpha_d^D} \rfloor$ of its children, a number which depends on the *progressive widening constant* α_d^D. This is the so-called progressive widening trick [10].

For a random node z, we actually have the same property, depending on the *double progressive widening constant* α_d^R: this is the so-called double progressive widening trick ([7]; see also [11]).

4 Main Result

Definition 2 (Exponentially sure in n). *We say that some property (P) depending on an integer n is* exponentially sure in n *(denoted e.s.) if there exists positive constants C, h, η such that the probability that (P) holds is at least*

$$1 - C\exp(-hn^{\eta}).$$

Theorem 1. *Define all exploration coefficients e_d and all progressive widening coefficients α_d as in Table 4. There is a constant $C > 0$, only depending on $d_{\max}$, such that after n episodes of PUCT, for every node z at depth d we have*

$$|\hat{V}(z) - V^*(z)| \leq \frac{C}{n(z)^{\gamma_d}} \text{ e.s. in } n(z) \tag{5}$$

Additionally, for every node $w = (z, a)$ at depth $d + \frac{1}{2}$ we have

$$|\hat{V}(w) - V^*(w)| \leq \frac{C}{n(w)^{\gamma_{d+\frac{1}{2}}}} \text{ e.s. in } n(w) \tag{6}$$

Corollary 1. *After n episodes, let $w_n(r)$ be the most simulated child node of r. Then,*

$$w_n(r) \text{ is optimal with precision } O\left(n^{-\frac{1}{10d_{\max}}}\right) \text{ e.s. in } n \tag{7}$$

Table 1. Definition of coefficients and convergence rates

Decision Node (d integer)	Random Node (d semi-integer)
$\alpha_d^D := \frac{1}{10(d_{\max} - d) - 3}$ for $d \leq d_{\max} - 1$ $e^d := \frac{1}{2p}\left(1 - \frac{3}{10(d_{\max} - d)}\right)$ for $d \leq d_{\max} - 1$	$\alpha_d^R := \begin{cases} \frac{3}{10(d_{\max}-d)-3} & \text{for } d \leq d_{\max} - \frac{3}{2} \\ 1 & \text{for } d = d_{\max} - \frac{1}{2} \end{cases}$
$\gamma_d^D := \frac{1}{10(d_{\max} - d)}$ for $d \leq d_{\max} - 1$	$\gamma_d^R := \frac{1}{10(d_{\max} - d) - 2}$ for $d \leq d_{\max} - \frac{1}{2}$

The proof is based on an induction on the following property and is detailed in the following three sections. Let us define this property.

Definition 3 (Induction property $Cons(\gamma_d, d)$).
There is a $C_d > 0$ such that for all nodes at integer depth d,

$$|\hat{V}(z) - V^*(z)| \leq C_d n(z)^{-\gamma_d} \textit{ e.s. in } n(z)$$

and for all nodes w at semi integer depths $d + \frac{1}{2}$,

$$|\hat{V}(w) - V^*(w)| \leq C_{d+\frac{1}{2}} n(w)^{-\gamma_{d+\frac{1}{2}}} \textit{ e.s. in } n(w)$$

In Section 5, we show that if $Cons(\gamma_d, d)$ holds for $d \geq 1$, i.e. for decision nodes in one given layer, then $Cons(\gamma_{d-\frac{1}{2}}, d - \frac{1}{2})$ holds, i.e. holds for the random nodes in the above layer. In Section 6, we show that if $Cons(\gamma_{d+\frac{1}{2}}, d + \frac{1}{2})$ holds for $d \geq 0$, i.e. for random nodes in one given layer, then $Cons(\gamma_d, d)$ holds, i.e. holds for the decision nodes in the above layer. Finally, we establish in Section 7 that $Cons(\gamma, d)$ holds for maximal depth d_{max}, which will settle the proof of Theorem 5.

5 From Decision Nodes to Random Nodes

In this section we consider a random node w with semi-integer depth $d - \frac{1}{2} \geq 0$. We suppose that there exist a $\gamma_d^D > 0$ such that $Cons(\gamma_d^D, d)$ holds for any child node z of w. Recall that all nodes at this depth have $\lfloor n^{\alpha^R_{d-\frac{1}{2}}} \rfloor$ constructed children when they have been visited n times. We will show that we can define $\alpha^R_{d-\frac{1}{2}}$ so that $Cons(\gamma^R_{d-\frac{1}{2}}, d - \frac{1}{2})$ holds. For convenience, if w is a random node, we will refer to the i^{th} child z_i of w by its index i directly. Then, the number of visits in z_i after the n^{th} iteration of PUCT will be simply called $n(i)$ instead of $n(z_i)$. Similarly, the empirical value of this node will be noted $\hat{V}(i)$ instead of $\hat{V}(z_i)$.

5.1 Children of Random Nodes are Selected almost the Same Number of Times

With our politics for dealing with random nodes, described in section 3, the k^{th} child of a random node w is constructed at episode $\lceil k^{\frac{1}{\alpha}} \rceil$. We now show that all constructed children of w but the last one are visited almost the same number of times.

Lemma 1. *Let w be a random node with progressive widening coefficient $\alpha \in]0; 1[$. Then after the n^{th} visit of w in the simulation, all children z_i, z_j of w with $1 \leq i, j < \lfloor n^\alpha \rfloor$ satisfy*

$$|n(i) - n(j)| \leq 1. \qquad (8)$$

In fact, in the next section, we will only use the following consequence of Lemma 1.

Corollary 2. *When a random node z is visited for the n^{th} time, all children of z have been selected at most $\frac{n}{\lfloor n^\alpha \rfloor - 1}$ times, and all children of z but the last one have been selected at least $\frac{n}{\lfloor n^\alpha \rfloor} - 1$ times.*

Proof. For length reasons, we only provide the following sketch of the proof:

Let us write k the k^{th} child of w for all $k \geq 1$, and $n_k = \lceil k^{\frac{1}{\alpha}} \rceil$ the number of visits in w when child k was introduced. Remark the statement of Lemma 1 is equivalent to:

(8) is satisfied *for all children* of z at every time step $n_k - 1$ for $k \geq 2$.
Then, prove the above statement by induction, by proving the following equivalent formulation: $n_{k+1} - n_k \geq \lfloor \frac{n_k - 1}{k-1} - 1 \rfloor$. □

5.2 Consistency of Random Nodes

Lemma 2 (Random nodes are consistent). . *If there is a $1 \geq \gamma_d > 0$ such that for any child z of the random node w we have $Cons(\gamma_d, d)$, then we have $Cons(\gamma_{d-\frac{1}{2}}, d - \frac{1}{2})$, with $\gamma_{d-\frac{1}{2}} = \frac{\gamma_d}{1+3\gamma_d}$ if we define the progressive widening coefficient $\alpha^R_{d-\frac{1}{2}}$ by $\alpha^R_{d-\frac{1}{2}} = \frac{3\gamma_d}{1+3\gamma_d}$.*

Proof. From now on, w is fixed in order to simplify notation; therefore, we simply denote $\alpha^R_{d-\frac{1}{2}}$ by α, and $n(w)$ by n.

Fix n such that $n^\alpha \geq 3$. Define $i_0 = \lfloor n^\alpha \rfloor$ as the last constructed child of node w, and $r = \lfloor n^\alpha \rfloor - 1 = i_0 - 1$. To prove the result, we need to prove an upper bound on the following quantity, that holds exponentially surely in n:

$$|\hat{V}(w) - V^*(w)| = |\left(\sum_{1 \leq i < i_0} \frac{n(i)}{n} \hat{V}(i) + \frac{n(i_0)}{n} \hat{V}(i_0) \right) - V^*(w)|$$

Decompose this as

$$|\hat{V}(w) - V^*(w)| \leq |\sum_{1 \leq i < i_0} \left(\frac{n(i)}{n} - \frac{1}{r} \right) \hat{V}(i)| \tag{9}$$

$$+ |\sum_{1 \leq i < i_0} \frac{1}{r} \left(\hat{V}(i) - V^*(i) \right)| \tag{10}$$

$$+ |\sum_{1 \leq i < i_0} \frac{1}{r} \left(V^*(i) - V^*(w) \right)| \tag{11}$$

$$+ |\frac{n(i_0)}{n} \hat{V}(i_0)| \tag{12}$$

First consider (9). By Lemma 1, there is a integer p such that all children $i = 1, \cdots, i_0 - 1$ have been selected p or $p+1$ times, with $p = O\left(n^{1-\alpha}\right)$. So, we have for all $i = 1, 2, \cdots, i_0 - 1$,

$$|\frac{n(i)}{n} - \frac{1}{\lfloor n^\alpha \rfloor - 1}| \leq |\frac{p}{n} - \frac{1}{\lfloor n^\alpha \rfloor - 1}| + \frac{1}{n}$$

The definition of p gives $(i_0 - 1)p \leq n \leq i_0(p+1)$, so that

$$|\frac{p}{n} - \frac{1}{i_0 - 1}| \leq \frac{i_0 + p}{(i_0 - 1)n} = O\left(\frac{1}{n} + \frac{1}{n^{2\alpha}}\right)$$

so that in the end for (9) we have

$$|\sum_{1 \leq i < i_0} \left(\frac{n(i)}{n} - \frac{1}{r}\right) \hat{V}(i)| = O\left(n^{\alpha}\left(\frac{1}{n} + \frac{1}{n^{2\alpha}} + \frac{1}{n}\right)\right) = O\left(\frac{1}{n^{1-\alpha}} + \frac{1}{n^{\alpha}}\right)$$

Consider now (10). $Cons(\gamma_d, d)$ holds, so for each child $i = 1, 2, \cdots, \lfloor n^{\alpha} \rfloor - 1$ of w, Lemma 1 leads to:

$$|\hat{V}(i) - V(i)| \leq C_d p^{-\gamma_d} \leq C_d \frac{1}{\lfloor n^{1-\alpha} \rfloor^{\gamma_d}} \text{ e.s. in } n^{1-\alpha}$$

Finally for (10) it is exponentially sure in n that

$$|\sum_{0 \leq i < i_0} \frac{1}{\lfloor n^{\alpha} \rfloor - 1} \left(\hat{V}(i) - V^*(i)\right)| = O\left(\frac{1}{n^{(1-\alpha)\gamma_d}}\right). \tag{13}$$

Now we turn to (11). Since w is a random node, the value $V^*(i)$ of each new child i of w constructed by the algorithm is given by a random law whose mean is $V^*(w)$. Thus we can apply Hoeffding's inequality to the sum in (11) and we obtain that for $t > 0$,

$$|\sum_{0 \leq i < i_0} \frac{1}{\lfloor n^{\alpha} \rfloor - 1} \left(V^*(i) - V^*(w)\right)| \leq t \tag{14}$$

with probability at least $1 - 2\exp\left(-2t^2\left(\lfloor n^{\alpha} \rfloor - 1\right)\right) = 1 - 2\exp(-Cn^{\frac{\gamma_d}{1+3\gamma_d}})$

with $t := n^{-\frac{\gamma'}{1+3\gamma_d}}, \alpha = \frac{3\gamma_d}{1+3\gamma_d},$ and $C > 0$. This proves that (14) is e.s. in n.

Finally consider (12): since the last child of w has been selected at most p times, we have

$$\left|\frac{n(i_0)}{n}\hat{V}(i_0)\right| = \frac{1}{n} \times O\left(\frac{n}{n^{\alpha}}\right) = O\left(\frac{1}{n^{\alpha}}\right).$$

All in all, we have have shown that it is exponentially sure in $n = n(w)$ that

$$|\hat{V}(w) - V^*(w)| = O\left(\underbrace{\frac{1}{n^{1-\alpha}} + \frac{1}{n^{\alpha}}}_{(9)} + \underbrace{\frac{1}{n^{(1-\alpha)\gamma_d}}}_{(10)} + \underbrace{\frac{1}{n^{\frac{\gamma_d}{1+3\gamma_d}}}}_{(11)} + \underbrace{\frac{1}{n^{\alpha}}}_{(12)}\right). \tag{15}$$

With $\alpha = \frac{3\gamma_d}{1+3\gamma_d}$ and $\gamma_d \leq 1$, it is straightforward to check that the smallest exponent is $\frac{\gamma_d}{1+3\gamma_d}$, so that $Cons(\gamma_{d-\frac{1}{2}}, d - \frac{1}{2})$ is true with $\gamma_{d-\frac{1}{2}} = \frac{\gamma_d}{1+3\gamma_d}\Box$

6 From Random Nodes to Decision Nodes

Let z be a non leaf decision node at depth d. In this section, we will show that if the induction property holds for all random nodes at depth $d + \frac{1}{2}$, it will hold for z.

Lemma 3 (Children of decision nodes are selected infinitely often). *Let f be a non-decreasing map from $\mathbb{N}$ to $\mathbb{N}$. Consider a stochastic bandit setting with a countable set of children, progressive widening coefficient α and exploration function f, i.e. the score at time n of a child i is computed by*

$$sc_n(i) = \hat{V}_n(i) + \sqrt{\frac{f(n)}{n(i)}}.$$

Then if i denotes the i^{th} constructed child, for all $n \geq i^{\frac{1}{\alpha(1-\alpha)}}$ we have

$$n(i) \geq \frac{1}{4} \min(f(n^{1-\alpha}), n^{1-\alpha}).$$

In particular, all constructed children are selected infinitely often provided that $\lim_{+\infty} f = +\infty$.

Proof. Fix n and consider the child i_0 maximizing $n(i_0)$, i.e. the most selected child at time n. Let n' be the last time i_0 has been selected. Since there are at most n^α children at time n we have

$$n'(i_0) = n(i_0) \geq \frac{n}{n^\alpha} = n^{1-\alpha} \tag{16}$$

where (i) $n'(i_0)$ is the number of times i_0 has been drawn before time n'; (ii) $n(i_0)$ is the number of times i_0 has been drawn before time n. Thus we also have

$$n' \geq n'(i_0) \geq n^{1-\alpha}. \tag{17}$$

Consider now any child i already constructed at time n'. Since i_0 was selected at time n' we must have

$$\sqrt{\frac{f(n')}{n'(i)}} \leq sc_{n'}(i) \leq sc_{n'}(i_0) \leq 1 + \sqrt{\frac{f(n')}{n'(i_0)}}. \tag{18}$$

Rewriting 18 and using 16 leads to

$$\frac{1}{\sqrt{n'(i)}} \leq \frac{1}{\sqrt{n^{1-\alpha}}} + \frac{1}{\sqrt{f(n')}} \leq \frac{2}{\sqrt{\min(f(n'), n^{1-\alpha})}} \tag{19}$$

so that for all children i at time n existing at time n' we have

$$n(i) \geq n'(i) \geq \frac{1}{4} \min\left(f(n^{1-\alpha}), n^{1-\alpha}\right)$$

as announced. Finally, note that a child i existed at time n' if $i \leq (n^{1-\alpha})^\alpha \leq n'^\alpha$, which leads to the prescribed condition. □

Corollary 3. *For the exploration function $f(n) = n^e$ with $0 < e < 1$ we obtain*

$$n(i) \geq \frac{1}{4} n^{e(1-\alpha)} \text{ if } i \leq n^{\alpha(1-\alpha)}.$$

Lemma 4 (Decision nodes are consistent). . *If there is a $\frac{1}{2} > \gamma_{d+\frac{1}{2}} > 0$ such that for any child w of the decision node z we have $Cons(\gamma_{d+\frac{1}{2}}, d+\frac{1}{2})$, then we have $Cons(\gamma_d, d)$ with $\gamma_d = \frac{\gamma_{d+\frac{1}{2}}}{1+7\gamma_{d+\frac{1}{2}}}$ if we define the progressive widening coefficient α_d^D by $\alpha_d^D = \frac{\gamma_{d+\frac{1}{2}}}{1+4\gamma_{d+\frac{1}{2}}}$.*

Proof. Let z be a decision node at depth $d \geq 0$. For simplicity, we note $\alpha_d = \alpha$ and $e_d = e$. Suppose that there is a $\frac{1}{2} > \gamma_{d+\frac{1}{2}} > 0$ such that for all random nodes w at depth $d+\frac{1}{2}$, $Cons(\gamma_{d+\frac{1}{2}}, d+\frac{1}{2})$ is true. To show $Cons(\gamma_d, d)$, we will proceed in two steps: first we establish an upper bound on $\hat{V}(z) - V^*(z)$, and then a lower bound.

Upper bound. First we obtain an upper bound on $\hat{V}(z) - V^*(z)$. Let $\epsilon < 1-\alpha$ to be fixed later. We partition the children of z in two classes:

- class I : children i such that $n(i) \leq n(z)^{1-\alpha-\epsilon}$;
- class II : other children;

$$\begin{aligned}
\hat{V}(z) - V^*(z) &= \sum_{i \text{ in class } I} \frac{n(i)}{n(z)}(\hat{V(i)} - V^*(z)) + \sum_{i \text{ in class } II} \frac{n(i)}{n(z)}(\hat{V(i)} - V^*(z)) \\
&\leq \sum_{i \text{ in class } I} \frac{n(i)}{n(z)} + \sum_{i \text{ in class } II} \frac{n(i)}{n(z)}(\hat{V}(i) - V^*(i)) \\
&\leq \frac{n^\alpha \times n^{1-\alpha-\epsilon}}{n} + C_{d+\frac{1}{2}}(n)^{-\gamma_{d+\frac{1}{2}}(1-\alpha-\epsilon)} \text{e.s. in } n^{-\gamma_{d+\frac{1}{2}}(1-\alpha-\epsilon)} \text{ by induction} \\
&\leq n^{-\epsilon} + C_{d+\frac{1}{2}} n^{-\gamma_{d+\frac{1}{2}}(1-\alpha-\epsilon)}.
\end{aligned}$$

We now choose $\epsilon = \frac{\gamma_{d+\frac{1}{2}}(1-\alpha)}{1+\gamma_{d+\frac{1}{2}}}$ and obtain

$$\hat{V}(z) - V(z) \leq (1 + C_{d+\frac{1}{2}}) n^{-\gamma_{d+\frac{1}{2}} \frac{1-\alpha}{1+\gamma_{d+\frac{1}{2}}}} \text{ e.s. in } n \tag{20}$$

Lower bound.

We assumed that there exists a constant θ such that when we pick a new child for z, it has a value satisfying $V(i) \geq V^*(z) - \Delta$ with probability at least $\min(1, \theta\Delta^p)$.

The induction hypothesis on the next level gives us a fixed coefficient $\gamma_{d+\frac{1}{2}} \in]0; 0.5[$ such that all children w of z verify e.s. in $n(w)$:

$$\left| V^*(w) - \hat{V}(w) \right| \leq C_{d+\frac{1}{2}} n(w)^{-\gamma_{d+\frac{1}{2}}}.$$

The parameters to be fixed on this level are

- the progressive widening coefficient $\alpha := \frac{\gamma_{d+\frac{1}{2}}}{1+4\gamma_{d+\frac{1}{2}}}$;
- the exploration coefficient $e := \frac{1}{1+4\gamma_{d+\frac{1}{2}}} - \frac{1}{\gamma_{d+\frac{1}{2}}}(1-\frac{1}{2p})\alpha = \frac{1}{2p(1+4\gamma_{d+\frac{1}{2}})}$.

To these coefficients we add a parameter ξ which we define by

$$\xi := \frac{1}{1+e\gamma_{d+\frac{1}{2}}(1-\alpha)} \tag{21}$$

$$\text{and let} \quad \varDelta := \left(\frac{1}{4}n^{\xi e(1-\alpha)}\right)^{-\gamma_{d+\frac{1}{2}}}. \tag{22}$$

First step : exponentially surely in n there exists at time $\lceil n^{\xi(1-\alpha)}\rceil$ a child i_0 of z such that

$$V(i_0) \geq V(z) - \varDelta \textbf{ and } i_0 \leq n^{\xi(1-\alpha)\alpha}. \tag{23}$$

At time step $\lceil n^{\xi(1-\alpha)}\rceil$, the number of children of z is a at least $\lfloor n^{\xi(1-\alpha)\alpha}\rfloor$. The (true hidden optimal) values of these children being given randomly and independently, the probability there is not a single child i_0 with $V(i_0) \geq V^*(z) - \varDelta$ at time $\lceil n^{\xi(1-\alpha)}\rceil$ is at most

$$p_n := (1-\theta\varDelta^p)^{\lfloor n^{\xi(1-\alpha)\alpha}\rfloor}$$

$$\begin{aligned}
\log p_n &\sim_n n^{\xi(1-\alpha)\alpha}\log(1-\theta\varDelta^p)\\
&\sim_n -n^{\xi(1-\alpha)\alpha}\theta\left(\frac{1}{4}n^{\xi e(1-\alpha)}\right)^{-\gamma_{d+\frac{1}{2}}p}\\
&\sim_n -4^{\gamma_{d+\frac{1}{2}}p}\theta n^{\xi(1-\alpha)(\alpha-e\gamma_{d+\frac{1}{2}}p)}\\
&\sim_n -4^{\gamma_{d+\frac{1}{2}}p}\theta n^{\xi(1-\alpha)0.5\alpha}.
\end{aligned}$$

The exponent of n in this quantity being positive, we deduce that the existence of i_0 is exponentially sure in n.

Second step: e.s. in n, all children selected at a time n' between n^ξ and n have a high score.

Let n' be such that $n^\xi \leq n' \leq n$. Then ${n'}^{\alpha(1-\alpha)} \geq n^{\xi(1-\alpha)\alpha} \geq i_0$. And, by Corollary 3,

$$n'(i_0) \geq \frac{1}{4}{n'}^{e(1-\alpha)} \geq \frac{1}{4}n^{\xi e(1-\alpha)}.$$

Hence there exists a $C' > 0$ by the induction hypothesis such that we have, as long as $n^\xi \leq n' \leq n$,

$$\begin{aligned}
\hat{V}(i_0) &\geq V^*(i_0) - C'\left(\frac{1}{4}n^{\xi e(1-\alpha)}\right)^{-\gamma_{d+\frac{1}{2}}} \text{ e.s. in } n'\\
&\geq V^*(z) - (1+C')\varDelta \text{ e.s. in } n'.
\end{aligned}$$

Consider any child i_1 chosen by the algorithm at a time $n' \geq n^{\xi}$, i.e. the one which has the greatest score at time n'. All values being considered at time n', we have

$$\hat{V}(i_1) + \sqrt{\frac{{n'}^e}{n'(i_1)}} \geq \hat{V}(i_0) + \sqrt{\frac{{n'}^e}{n'(i_0)}},$$

$$\text{hence } \hat{V}(i_1) + \sqrt{\frac{n^e}{n'(i_1)}} \geq V^*(z) - (1 + C')\Delta \text{ e.s. in } n'. \tag{24}$$

To conclude this part, all we have to do is to show that some property exponentially sure in n' is also exponentially sure in n. This easily follows from the fact that $n' \geq n^{\xi}$ and that ξ, is bounded below by some constant. One can easily check from the definition of ξ that $\xi \geq \frac{2}{3}$, since $e \leq \frac{1}{2}$.

Third step : lower bound on $\hat{V}(z)$.

Consider a child i_1 selected after n^{ξ}. By the previous step, exponentially surely in n, this child must either satisfy

$$\sqrt{\frac{n^e}{n(i_1)}} \geq \Delta \tag{25}$$

$$\text{or } \hat{V}(i_1) \geq V(z) - (2 + C')\Delta. \tag{26}$$

Under this hypothesis we can split the children of z in three categories:

1. children i_1 visited only before time n^{ξ} ;
2. children i_1 visited after n^{ξ} satisfying (25) ;
3. children i_1 visited after n^{ξ} satisfying (26) .

Let us use this decomposition to lower bound the sum

$$\hat{V}(z) - V^*(z) = \sum_{i=1\cdots\lfloor n^{\alpha}\rfloor} \frac{n(i)}{n}(\hat{V}(i) - V^*(z)).$$

For the children in the first category, we have

$$\left| \sum_{i_1 \text{in cat.1}} \frac{n(i_1)}{n}(\hat{V}(i_1) - V^*(z)) \right| \leq \frac{\sum_{i_1 \text{in cat.1}} n(i_1)}{n} \leq \frac{n^{\xi}}{n}.$$

For children in the second category, since there are at most n^{α} of these children, we have

$$\left| \sum_{i_1 \text{in cat.2}} \frac{n(i_1)}{n}(\hat{V}(i_1) - V^*(z)) \right| \leq \frac{\sum_{i_1 \text{in cat.2}} n(i_1)}{n} \leq \frac{n^{\alpha}}{n}\frac{n^e}{\Delta^2} = \frac{n^{\alpha+e-1}}{\Delta^2}.$$

Finally, using (26) for the third category of children, we see that

$$\hat{V}(z) - V^*(z) \geq -(2 + C')\Delta(1 - n^{\xi-1}) - n^{\xi-1} - \frac{n^{\alpha+e-1}}{\Delta^2}.$$

Now we compare the three terms

$$\mathit{\Delta}, n^{\xi-1} \text{ and } \frac{n^{\alpha+e-1}}{\mathit{\Delta}^2}. \tag{27}$$

By (21) we have $\xi - 1 = -\xi e \gamma_{d+\frac{1}{2}}(1-\alpha)$, thus by (22), $n^{\xi-1} = 4^{-\gamma_{d+\frac{1}{2}}} \mathit{\Delta} \leq \mathit{\Delta}$.

This implies that the term $n^{\xi-1}$ in the three terms (Eq. 27) is $O(\mathit{\Delta})$. We now compare the two other terms; from the definition of $\mathit{\Delta}$, we see that we must compare $n^{\alpha+e-1}$ and $\mathit{\Delta}^3 = 4^{3\gamma_{d+\frac{1}{2}}} n^{-3\xi e(1-\alpha)\gamma_{d+\frac{1}{2}}}$. Using the definitions of ξ, e and α, one can check that:

$$1 - e - \alpha \geq \frac{3\gamma_{d+\frac{1}{2}} + \frac{1}{2}}{1 + 4\gamma_{d+\frac{1}{2}}} \geq \frac{1}{2}$$

and, using $\xi \leq 1$, $(1-\alpha) \leq 1$, $e\gamma_{d+\frac{1}{2}} = \frac{\gamma_{d+\frac{1}{2}}}{2(1+4\gamma_{d+\frac{1}{2}})} \leq \frac{1}{8}$,

$$3\xi e(1-\alpha)\gamma_{d+\frac{1}{2}} \leq \frac{3}{8}$$

$$n^{e+\alpha-1} \leq n^{-3\xi e(1-\alpha)\gamma_{d+\frac{1}{2}}} = 4^{-3\gamma_{d+\frac{1}{2}}} \mathit{\Delta}^3 \leq \mathit{\Delta}^3$$

so that $\hat{V}(z) - V(z) \geq -(5 + C')\mathit{\Delta}$. Finally, one can check that $\xi e(1-a)\gamma_{d+\frac{1}{2}} \geq \frac{\gamma_{d+\frac{1}{2}}}{1+7\gamma_{d+\frac{1}{2}}}$ so that $\hat{V}(z) - V^*(z) \geq -(5 + C')4^{\gamma_{d+\frac{1}{2}}} n^{\frac{\gamma_{d+\frac{1}{2}}}{1+7\gamma_{d+\frac{1}{2}}}}$ which can now be written $\hat{V}(z) - V^*(z) \geq -Cn^{-\gamma}$ with $C := (5 + C')4^{\gamma_{d+\frac{1}{2}}}$ and $\gamma = \frac{\gamma_{d+\frac{1}{2}}}{1+7\gamma_{d+\frac{1}{2}}}$.□

7 Base Step, Initialization and Conclusion of the Proof

Let w be a random node of depth $d_{\max} - \frac{1}{2}$. Its children are leaf nodes, and all have a fixed reward in $[0;1]$. These children form a ensemble of independent and identically distributed variables, all following the random distribution associated with w, of mean $V^*(w)$. Hoeffding's inequality gives, for $t > 0$,

$$\mathbb{P}\left(|\frac{1}{n} \sum_{z_i child of w} V^*(z_i) - V^*(w)| \geq t\right) \leq 2\exp(-2t^2 n).$$

Setting the exploration coefficient $\alpha_{d_{\max}-\frac{1}{2}}$ to 1 (since there is no point in selecting again children with a constant reward) and t to $n^{-\frac{1}{3}}$, we obtain

$$\mathbb{P}\left(|V\hat{(w)} - V^*(w)| \geq n^{-\frac{1}{3}}\right) \leq 2\exp(-2n^{\frac{1}{3}})$$

so that $|V\hat{(w)} - V^*(w)| \leq n^{-\frac{1}{3}}$ is exponentially sure in n, i.e. $Cons(\frac{1}{3}, d_{\max} - \frac{1}{2})$ holds. Of course one can consider a coefficient different from $\frac{1}{3}$ for t, as long as it is less than $\frac{1}{2}$ – we just aim so as to simplify the definition of coefficients. This gives a singular value of $\alpha^R_{d_{\max}-\frac{1}{2}} = 1$ and an initialization of the convergence rate as $\gamma^R_{d_{\max}-\frac{1}{2}} = \frac{1}{3}$. It is now elementary to check this value of $\frac{1}{3}$ for γ at depth $d_{\max} - \frac{1}{2}$, together with recursive definitions of coefficients derived in Lemmas 2 and 4, yield the values given on Table 4. This concludes the proof of Theorem 5.

8 Experimental Validation

In this section, we show some experimental results, by implementing PUCT on two tests problems. We used fixed parameters α and e, quickly tuned by hand. We added a custom default policy, as seen in [6] and [8], that is computed offline using Direct Policy Search (DPS), once per problem instance. We also gave heavier weights to the decisions with high average value when computing the empirical value of a state, as it showed increased performances in practice. There are many ways to finely tune PUCT that we did not explore. Our goal was simply to check that our PUCT has a satisfying behaviour, to verify our theoretical results. We acknowledge that depending on implementation subtleties, results can vary. Our source code is available upon request.

Cart pole. We used the well known benchmark of cart pole, and more precisely the version presented in [17]. As our code uses time budget, and not a limit in the number of iterations, we only approximated their limit of 200 roll outs (on our machine, 0.001 second per action. We took HOLOP as a baseline, that yields an average reward of -47.45[17]. Our results are shown in Table 2. Though cart pole is not as challenging as real world applications, these results are encouraging and supporting our theoretical results of consistency.

Unit commitment. We used a unit commitment problem, inspired by ongoing work with an industrial partner. The agent owns 2 water reservoirs and 5 power plants. Each reservoir is a free but limited source of energy. Each power plant has a fixed capacity, has a fixed cost to be turned on, as well as quadratic running costs that change over time. The time horizon was fixed to 6 time steps. At each time step, the agent decides how to produce energy in order to satisfy a varying demand, and the water reservoirs receive a random inflow. Failure to satisfy the demand incurs a prohibitive cost. This problem is challenging for many reasons, including: the action space is non convex, the objective function is non linear and discontinuous, there are binary and continuous variables, and finally, the action space can be subject to operational constraints that make a discretization by hand very tedious. The purpose of PUCT in this application is not to solve all of it, but rather to improve existing solvers. This is an especially promising method, with the many powerful heuristics available for this problem. The results are shown in Table 2. PUCT manages to reliably improve the actions suggested by DPS, and its performances increase with the time budget it is given.

Table 2. Left: Cart Pole results; episodes are 200 time steps long. Right: Unit Commitment results, with 2 stocks, 5 plants, and 6 time steps.

Budget (s)	0.001	0.004
HOLOP	-47.45 ± .	.
DPS	-838.7 ± 78.0	-511.0 ± 100.0
PUCT+DPS	-13.84 ± 0.80	-11.11 ± 0.95

Budget (s)	0.04	0.16	0.64
DPS	-8.02 ± 0.98	-7.06 ± 0.024	-6.98 ± 0.03
PUCT+DPS	-7.23 ± 0.45	-6.69± 0.03	6.57 ± 0.02

9 Conclusion

[13] have shown the consistency of the UCT approach for finite Markov Decision Processes. We have shown the consistency of our modified version, with polynomial exploration and double progressive widening, for a more general case. [7] have shown that the classical UCT is not consistent in this case and already proposed double progressive widening; we here give a proof of the consistency of this approach, when we use polynomial exploration; [7] was using logarithmic exploration.

Some extensions of our work are straightforward. We considered trees, but the extension to MDP with possibly two distinct paths leading to the same node is straightforward. Also, we assumed, only for simplifying notation, that the probability that a random node leads twice to the same decision node (when drawn independently with the probability distribution of the random node) is zero, but the extension is possible. On the other hand, we point out two deeper limitations of our work: (i) We do not know if similar results can be derived without switching to polynomial exploration. (ii) The general case of a possibly cyclic MDP with unbounded horizon is not covered by our result.

We have shown consistency in the sense that Bellman values are properly estimated. This does not explain which decision should be actually made when PUCT has been performed for generating episodes and estimating V values. Our result implies that choosing the action by empirical distribution of play (i.e. randomly draw a decision with probability equal to the frequency at which it was simulated during episodes; see discussion in [4]) is asymptotically consistent. Also, choosing the most simulated child is consistent (this is a classical method in UCT), as well as selecting the child with best $\hat{V}$ among child nodes of the root of class II; our results do not show the superiority of one or another of these recommendation methodologies.

Our experimental results on the classical Cart pole problem show that PUCT outperforms HOLOP; PUCT also outperformed a specialized DPS on a unit commitment problem. This last empirical result is especially interesting because unit commitment problems are, in practice, highly non Markovian. And, even though we worked in the framework of MDP to relate to its abundant literature, our algorithm does not actually need the random process to be Markovian, as the history is naturally embedded in the tree structure. Hence, PUCT could be a way to approach difficult and more general non Markovian continuous sequential decision making problems.

References

1. P. Auer, R. Ortner, and C. Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In N. H. Bshouty and C. Gentile, editors, *COLT*, volume 4539 of *Lecture Notes in Computer Science*, pages 454–468. Springer, 2007.
2. R. Bellman. *Dynamic Programming.* Princeton Univ. Press, 1957.
3. D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. 28(1):52 – 63, 2013.
4. A. Bourki, M. Coulm, P. Rolet, O. Teytaud, and P. Vayssière. Parameter Tuning by Simple Regret Algorithms and Multiple Simultaneous Hypothesis Testing. In *ICINCO2010*, page 10, funchal madeira, Portugal, 2010.
5. S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. Online optimization in x-armed bandits. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *NIPS*, pages 201–208. Curran Associates, Inc., 2008.
6. O. Buffet, C. Lee, W. Lin, and O. Teytaud. Optimistic heuristics for minesweeper. In *International Computer Symposium*, page 9, 2012.
7. A. Couetoux, J.-B. Hoock, N. Sokolovska, O. Teytaud, and N. Bonnard. Continuous Upper Confidence Trees. In *LION'11: Proceedings of the 5th International Conference on Learning and Intelligent OptimizatioN*, page TBA, Italie, Jan. 2011.
8. A. Couetoux, O. Teytaud, and H. Doghmen. Learning a move-generator for upper confidence trees. In R.-S. Chang, L. C. Jain, and S.-L. Peng, editors, *Advances in Intelligent Systems and Applications - Volume 1*, volume 20 of *Smart Innovation, Systems and Technologies*, pages 209–218. Springer Berlin Heidelberg, 2013.
9. R. Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. *In P. Ciancarini and H. J. van den Herik, editors, Proceedings of the 5th International Conference on Computers and Games, Turin, Italy*, pages 72–83, 2006.
10. R. Coulom. Computing elo ratings of move patterns in the game of go. In *Computer Games Workshop, Amsterdam, The Netherlands*, 2007.
11. A. Gerevini, A. E. Howe, A. Cesta, and I. Refanidis, editors. *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19-23, 2009.* AAAI, 2009.
12. R. D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *NIPS*, 2004.
13. L. Kocsis and C. Szepesvari. Bandit based Monte-Carlo planning. In *15th European Conference on Machine Learning (ECML)*, pages 282–293, 2006.
14. C.-S. Lee, M.-H. Wang, G. Chaslot, J.-B. Hoock, A. Rimmel, O. Teytaud, S.-R. Tsai, S.-C. Hsu, and T.-P. Hong. The Computational Intelligence of MoGo Revealed in Taiwan's Computer Go Tournaments. *IEEE Transactions on Computational Intelligence and AI in games*, 2009.
15. O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artif. Intell.*, 147(1-2):5–34, 2003.
16. C. R. Mansley, A. Weinstein, and M. L. Littman. Sample-based planning for continuous action markov decision processes. In F. Bacchus, C. Domshlak, S. Edelkamp, and M. Helmert, editors, *ICAPS*. AAAI, 2011.
17. A. Weinstein and M. L. Littman. Bandit-based planning and learning in continuous-action markov decision processes. In L. McCluskey, B. Williams, J. R. Silva, and B. Bonet, editors, *ICAPS*. AAAI, 2012.