Convex Regularization in Monte-Carlo Tree Search

Tuan Dam [1] **Carlo D'Eramo** [1] **Jan Peters** [1] **Joni Pajarinen** [1 2]

Abstract

Monte-Carlo planning and Reinforcement Learning (RL) are essential to sequential decision making. The recent AlphaGo and AlphaZero algorithms have shown how to successfully combine these two paradigms to solve large-scale sequential decision problems. These methodologies exploit a variant of the well-known UCT algorithm to trade off the exploitation of good actions and the exploration of unvisited states, but their empirical success comes at the cost of poor sample-efficiency and high computation time. In this paper, we overcome these limitations by introducing the use of convex regularization in Monte-Carlo Tree Search (MCTS) to drive exploration efficiently and to improve policy updates. First, we introduce a unifying theory on the use of generic convex regularizers in MCTS, deriving the first regret analysis of regularized MCTS and showing that it guarantees an exponential convergence rate. Second, we exploit our theoretical framework to introduce novel regularized backup operators for MCTS, based on the relative entropy of the policy update and, more importantly, on the Tsallis entropy of the policy, for which we prove superior theoretical guarantees. We empirically verify the consequence of our theoretical results on a toy problem. Finally, we show how our framework can easily be incorporated in AlphaGo and we empirically show the superiority of convex regularization, w.r.t. representative baselines, on well-known RL problems across several Atari games.

1. Introduction

Monte-Carlo Tree Search (MCTS) is a well-known algorithm to solve decision-making problems through the combination of Monte-Carlo planning and an incremental tree structure (Coulom, 2006). MCTS provides a principled approach for trading off between exploration and exploitation in sequential decision making. Moreover, recent advances have shown how to enable MCTS in continuous and large problems (Silver et al., 2016; Yee et al., 2016). Most remarkably, AlphaGo (Silver et al., 2016) and AlphaZero (Silver et al., 2017a;b) couple MCTS with neural networks trained using Reinforcement Learning (RL) (Sutton & Barto, 1998) methods, e.g., Deep Q-Learning (Mnih et al., 2015), to speed up learning of large scale problems. In particular, a neural network is used to compute value function estimates of states as a replacement of time-consuming Monte-Carlo rollouts, and another neural network is used to estimate policies as a probability prior for the therein introduced PUCT action selection strategy, a variant of well-known UCT sampling strategy commonly used in MCTS for exploration (Kocsis et al., 2006). Despite AlphaGo and AlphaZero achieving state-of-the-art performance in games with high branching factor like Go (Silver et al., 2016) and Chess (Silver et al., 2017a), both methods suffer from poor sample-efficiency, mostly due to the polynomial convergence rate of PUCT (Xiao et al., 2019). This problem, combined with the high computation time to evaluate the deep neural networks, significantly hinder the applicability of both methodologies.

In this paper, we provide a theory of the use of convex regularization in MCTS, which proved to be an efficient solution for driving exploration and stabilizing learning in RL (Schulman et al., 2015; 2017a; Haarnoja et al., 2018; Buesing et al., 2020). In particular, we show how a regularized objective function in MCTS can be seen as an instance of the Legendre-Fenchel transform, similar to previous findings on the use of duality in RL (Mensch & Blondel, 2018; Geist et al., 2019; Nachum & Dai, 2020a) and game theory (Shalev-Shwartz & Singer, 2006; Pavel, 2007). Establishing our theoretical framework, we can derive the first regret analysis of regularized MCTS, and prove that a generic convex regularizer guarantees an exponential convergence rate to the solution of the regularized objective function, which improves on the polynomial rate of PUCT. These results provide a theoretical ground for the use of arbitrary entropy-based regularizers in MCTS until now limited to maximum entropy (Xiao et al., 2019), among which we specifically study the relative entropy of policy

[1] Department of Computer Science, Technische Universität Darmstadt, Germany [2] Department of Electrical Engineering and Automation, Aalto University, Finland. Correspondence to: Tuan Dam <tuan.dam@tu-darmstadt.de>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

updates, drawing on similarities with trust-region and proximal methods in RL (Schulman et al., 2015; 2017b), and the Tsallis entropy, used for enforcing the learning of sparse policies (Lee et al., 2018). Moreover, we provide an empirical analysis of the toy problem introduced in Xiao et al. (2019) to evince the practical consequences of our theoretical results for each regularizer. Finally, we empirically evaluate the proposed operators in AlphaGo, on several Atari games, confirming the benefit of convex regularization in MCTS, and in particular the superiority of Tsallis entropy w.r.t. other regularizers.

2. Preliminaries

2.1. Markov Decision Processes

We consider the classical definition of a finite-horizon Markov Decision Process (MDP) as a 5-tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the finite discrete action space, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition kernel, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi \in \Pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a probability distribution of the event of executing an action a in a state s. A policy π induces a value function corresponding to the expected cumulative discounted reward collected by the agent when executing action a in state s, and following the policy π thereafter: $Q^\pi(s, a) \triangleq \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{i+k+1} | s_i = s, a_i = a, \pi\right]$, where r_{i+1} is the reward obtained after the i-th transition. An MDP is solved finding the optimal policy π^*, which is the policy that maximizes the expected cumulative discounted reward. The optimal policy satisfies the optimal Bellman equation (Bellman, 1954) $Q^*(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \max_{a'} Q^*(s', a')\right] ds'$, and is the fixed point of the optimal Bellman operator $\mathcal{T}^* Q(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \max_{a'} Q(s', a')\right] ds'$. We define the Bellman operator under the policy π as $\mathcal{T}_\pi Q(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \int_{\mathcal{A}} \pi(a'|s') Q(s', a') da'\right] ds'$, the optimal value function $V^*(s) \triangleq \max_{a \in \mathcal{A}} Q^*(s, a)$, and the value function under the policy π as $V^\pi(s) \triangleq \max_{a \in \mathcal{A}} Q^\pi(s, a)$.

2.2. Monte-Carlo Tree Search and Upper Confidence bounds for Trees

Monte-Carlo Tree Search (MCTS) is a planning strategy based on a combination of Monte-Carlo sampling and tree search to solve MDPs. MCTS builds a tree where the nodes are the visited states of the MDP, and the edges are the actions executed in each state. MCTS converges to the optimal policy (Kocsis et al., 2006; Xiao et al., 2019), iterating over a loop composed of four steps:

1. **Selection:** starting from the root node, a *tree-policy* is executed to navigate the tree until a node with unvisited children, i.e. expandable node, is reached;
2. **Expansion:** the reached node is expanded according to the tree policy;
3. **Simulation:** run a rollout, e.g. Monte-Carlo simulation, from the visited child of the current node to the end of the episode;
4. **Backup:** use the collected reward to update the action-values $Q(\cdot)$ of the nodes visited in the trajectory from the root node to the expanded node.

The tree-policy used to select the action to execute in each node needs to balance the use of already known good actions, and the visitation of unknown states. The Upper Confidence bounds for Trees (UCT) sampling strategy (Kocsis et al., 2006) extends the use of the well-known UCB1 sampling strategy for multi-armed bandits (Auer et al., 2002), to MCTS. Considering each node corresponding to a state $s \in \mathcal{S}$ as a different bandit problem, UCT selects an action $a \in \mathcal{A}$ applying an upper bound to the action-value function

$$\text{UCT}(s, a) = Q(s, a) + \epsilon \sqrt{\frac{\log N(s)}{N(s, a)}}, \tag{1}$$

where $N(s, a)$ is the number of executions of action a in state s, $N(s) = \sum_a N(s, a)$, and ϵ is a constant parameter to tune exploration. UCT asymptotically converges to the optimal action-value function Q^*, for all states and actions, with the probability of executing a suboptimal action at the root node approaching 0 with a polynomial rate $O(\frac{1}{t})$, for a simulation budget t (Kocsis et al., 2006; Xiao et al., 2019).

3. Regularized Monte-Carlo Tree Search

The success of RL methods based on entropy regularization comes from their ability to achieve state-of-the-art performance in decision making and control problems, while enjoying theoretical guarantees and ease of implementation (Haarnoja et al., 2018; Schulman et al., 2015; Lee et al., 2018). However, the use of entropy regularization is MCTS is still mostly unexplored, although its advantageous exploration and value function estimation would be desirable to reduce the detrimental effect of high-branching factor in AlphaGo and AlphaZero. To the best of our knowledge, the MENTS algorithm (Xiao et al., 2019) is the first and only method to combine MCTS and entropy regularization. In particular, MENTS uses a maximum entropy regularizer in AlphaGo, proving an exponential convergence rate to the solution of the respective softmax objective function and achieving state-of-the-art performance in some Atari games (Bellemare et al., 2013). In the following, motivated by the success in RL and the promising results of MENTS, we derive a unified theory of regularization in MCTS based on the Legendre-Fenchel transform (Geist et al., 2019), that generalizes the use of maximum entropy of MENTS to an arbitrary convex regularizer. Notably, our theoretical framework enables to rigorously motivate the advantages of using maximum entropy and other entropy-based regularizers, such as relative entropy or Tsallis entropy, drawing connections with their RL counterparts TRPO (Schulman et al., 2015) and Sparse DQN (Lee et al., 2018), as MENTS does with Soft Actor-Critic (SAC) (Haarnoja et al., 2018).

3.1. Legendre-Fenchel transform

Consider an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$, as previously defined. Let $\Omega : \Pi \rightarrow \mathbb{R}$ be a strongly convex function. For a policy $\pi_s = \pi(\cdot|s)$ and $Q_s = Q(s, \cdot) \in \mathbb{R}^{\mathcal{A}}$, the Legendre-Fenchel transform (or convex conjugate) of Ω is $\Omega^* : \mathbb{R}^{\mathcal{A}} \rightarrow \mathbb{R}$, defined as:

$$\Omega^*(Q_s) \triangleq \max_{\pi_s \in \Pi_s} \mathcal{T}_{\pi_s} Q_s - \tau \Omega(\pi_s), \tag{2}$$

where the temperature τ specifies the strength of regularization. Among the several properties of the Legendre-Fenchel transform, we use the following (Mensch & Blondel, 2018; Geist et al., 2019).

Proposition 1 *Let Ω be strongly convex.*

- *Unique maximizing argument: $\nabla\Omega^*$ is Lipschitz and satisfies*

$$\nabla\Omega^*(Q_s) = \arg\max_{\pi_s \in \Pi_s} \mathcal{T}_{\pi_s} Q_s - \tau\Omega(\pi_s). \tag{3}$$

- *Boundedness: if there are constants L_Ω and U_Ω such that for all $\pi_s \in \Pi_s$, we have $L_\Omega \leq \Omega(\pi_s) \leq U_\Omega$, then*

$$\max_{a \in \mathcal{A}} Q_s(a) - \tau U_\Omega \leq \Omega^*(Q_s) \leq \max_{a \in \mathcal{A}} Q_s(a) - \tau L_\Omega. \tag{4}$$

- *Contraction: for any $Q_1, Q_2 \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$*

$$\| \Omega^*(Q_1) - \Omega^*(Q_2) \|_\infty \leq \gamma \| Q_1 - Q_2 \|_\infty . \tag{5}$$

Note that if $\Omega(\cdot)$ is strongly convex, $\tau\Omega(\cdot)$ is also strongly convex; thus all the properties shown in Proposition 1 still hold[1].

Solving equation (2) leads to the solution of the optimal primal policy function $\nabla\Omega^*(\cdot)$. Since $\Omega(\cdot)$ is strongly convex, the dual function $\Omega^*(\cdot)$ is also convex. One can solve the optimization problem (2) in the dual space (Nachum & Dai, 2020b) as

$$\Omega(\pi_s) = \max_{Q_s \in \mathbb{R}^{\mathcal{A}}} \mathcal{T}_{\pi_s} Q_s - \tau\Omega^*(Q_s) \tag{6}$$

and find the solution of the optimal dual value function as $\Omega^*(\cdot)$. Note that the Legendre-Fenchel transform of the value conjugate function is the convex function Ω, i.e. $\Omega^{**} = \Omega$. In the next section, we leverage on this primal-dual connection based on the Legendre-Fenchel transform as both conjugate value function and policy function, to derive the regularized MCTS backup and tree policy.

3.2. Regularized backup and tree policy

In MCTS, each node of the tree represents a state $s \in \mathcal{S}$ and contains a visitation count $N(s, a)$. Given a trajectory, we define $n(s_T)$ as the leaf node corresponding to the reached state s_T. Let $s_0, a_0, s_1, a_1..., s_T$ be the state action trajectory in a simulation, where $n(s_T)$ is a leaf node of $\mathcal{T}$. Whenever a node $n(s_T)$ is expanded, the respective action values (Equation 7) are initialized as $Q_\Omega(s_T, a) = 0$, and $N(s_T, a) = 0$ for all $a \in \mathcal{A}$. For all nodes in the trajectory, the visitation count is updated by $N(s_t, a_t) = N(s_t, a_t)+1$, and the action-values by

$$Q_\Omega(s_t, a_t) = \begin{cases} r(s_t, a_t) + \gamma\rho & \text{if } t = T \\ r(s_t, a_t) + \gamma\Omega^*(Q_\Omega(s_{t+1})/\tau) & \text{if } t < T \end{cases} \tag{7}$$

where $Q_\Omega(s_{t+1}) \in \mathbb{R}^{\mathcal{A}}$ with $Q_\Omega(s_{t+1}, a), \forall a \in \mathcal{A}$, and ρ is an estimate returned from an evaluation function computed in s_T, e.g. a discounted cumulative reward averaged over multiple rollouts, or the value-function of node $n(s_{T+1})$ returned by a value-function approximator, e.g. a neural network pretrained with deep Q-learning (Mnih et al., 2015), as done in (Silver et al., 2016; Xiao et al., 2019). We revisit the E2W sampling strategy limited to maximum entropy regularization (Xiao et al., 2019) and, through the use of the convex conjugate in Equation (7), we derive a novel sampling strategy that generalizes to any convex regularizer

$$\pi_t(a_t|s_t) = (1 - \lambda_{s_t})\nabla\Omega^*(Q_\Omega(s_t)/\tau)(a_t) + \frac{\lambda_{s_t}}{|\mathcal{A}|}, \tag{8}$$

where $\lambda_{s_t} = \epsilon|\mathcal{A}|/\log(\sum_a N(s_t,a)+1)$ with $\epsilon > 0$ as an exploration parameter, and $\nabla\Omega^*$ depends on the measure in use (see Table 1 for maximum, relative, and Tsallis entropy). We call this sampling strategy *Extended Empirical Exponential Weight* (E3W) to highlight the extension of E2W from maximum entropy to a generic convex regularizer. E3W defines the connection to the duality representation using the Legendre-Fenchel transform, that is missing in E2W. Moreover, while the Legendre-Fenchel transform can be used to derive a theory of several state-of-the-art algorithms in RL, such as TRPO, SAC, A3C (Geist & Scherrer, 2011), our result is the first introducing the connection with MCTS.

[1] Other works use the same formula, e.g. Equation (2) in Niculae & Blondel (2017).

3.3. Convergence rate to regularized objective

We show that the regularized value V_Ω can be effectively estimated at the root state $s \in \mathcal{S}$, with the assumption that each node in the tree has a σ^2-subgaussian distribution. This result extends the analysis provided in (Xiao et al., 2019), which is limited to the use of maximum entropy.

Theorem 1 *At the root node s where $N(s)$ is the number of visitations, with $\epsilon > 0$, $V_\Omega(s)$ is the estimated value, with constant C and $\hat{C}$, we have*

$$\mathbb{P}(|V_\Omega(s) - V^*_\Omega(s)| > \epsilon) \leq C \exp\{\frac{-N(s)\epsilon}{\hat{C}\sigma \log^2(2 + N(s))}\}, \tag{9}$$

where $V_\Omega(s) = \Omega^*(Q_s)$ and $V^*_\Omega(s) = \Omega^*(Q^*_s)$.

From this theorem, we obtain that the convergence rate of choosing the best action a^* at the root node, when using the E3W strategy, is exponential.

Theorem 2 *Let a_t be the action returned by E3W at step t. For large enough t and constants $C, \hat{C}$*

$$\mathbb{P}(a_t \neq a^*) \leq Ct \exp\{-\frac{t}{\hat{C}\sigma(\log(t))^3}\}. \tag{10}$$

This result shows that, for every strongly convex regularizer, the convergence rate of choosing the best action at the root node is exponential, as already proven in the specific case of maximum entropy (Xiao et al., 2019).

4. Entropy-regularization backup operators

From the introduction of a unified view of generic strongly convex regularizers as backup operators in MCTS, we narrow the analysis to entropy-based regularizers. For each entropy function, Table 1 shows the Legendre-Fenchel transform and the maximizing argument, which can be respectively replaced in our backup operation (Equation 7) and sampling strategy E3W (Equation 8). Using maximum entropy retrieves the maximum entropy MCTS problem introduced in the MENTS algorithm (Xiao et al., 2019). This approach closely resembles the maximum entropy RL framework used to encourage exploration (Haarnoja et al., 2018; Schulman et al., 2017a). We introduce two novel MCTS algorithms based on the minimization of relative entropy of the policy update, inspired by trust-region (Schulman et al., 2015; Belousov & Peters, 2019) and proximal optimization methods (Schulman et al., 2017b) in RL, and on the maximization of Tsallis entropy, which has been more recently introduced in RL as an effective solution to enforce the learning of sparse policies (Lee et al., 2018). We call these algorithms RENTS and TENTS. Contrary to maximum and relative entropy, the definition of the Legendre-Fenchel and maximizing argument of Tsallis entropy is non-trivial, being

$$\Omega^*(Q_t) = \tau \cdot \text{spmax}(Q_t(s, \cdot)/\tau), \tag{11}$$

$$\nabla\Omega^*(Q_t) = \max\{\frac{Q_t(s,a)}{\tau} - \frac{\sum_{a \in \mathcal{K}} Q_t(s,a)/\tau - 1}{|\mathcal{K}|}, 0\}, \tag{12}$$

where spmax is defined for any function $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ as

$$\text{spmax}(f(s,\cdot)) \triangleq \sum_{a \in \mathcal{K}} \left(\frac{f(s,a)^2}{2} - \frac{(\sum_{a \in \mathcal{K}} f(s,a) - 1)^2}{2|\mathcal{K}|^2} \right) + \frac{1}{2}, \tag{13}$$

and $\mathcal{K}$ is the set of actions that satisfy $1 + if(s, a_i) > \sum_{j=1}^{i} f(s, a_j)$, with a_i indicating the action with the i-th largest value of $f(s,a)$ (Lee et al., 2018). We point out that the Tsallis entropy is not significantly more difficult to implement. Although introducing additional computation, requiring $O(|\mathcal{A}| \log(|\mathcal{A}|))$ time in the worst case, the order of Q-values does not change between rollouts, reducing the computational complexity in practice.

4.1. Regret analysis

At the root node, let each children node i be assigned with a random variable X_i, with mean value V_i, while the quantities related to the optimal branch are denoted by $*$, e.g. mean value V^*. At each timestep n, the mean value of variable X_i is V_{i_n}. The pseudo-regret (Coquelin & Munos, 2007) at the root node, at timestep n, is defined as $R^{\text{UCT}}_n = nV^* - \sum_{t=1}^{n} V_{i_t}$. Similarly, we define the regret of E3W at the root node of the tree as

$$\begin{aligned} R_n = nV^* - \sum_{t=1}^{n} V_{i_t} &= nV^* - \sum_{t=1}^{n} \mathbb{I}(i_t = i) V_{i_t} \\ &= nV^* - \sum_{i} V_i \sum_{t=1}^{n} \hat{\pi}_t(a_i|s), \end{aligned} \tag{14}$$

where $\hat{\pi}_t(\cdot)$ is the policy at time step t, and $\mathbb{I}(\cdot)$ is the indicator function.
The expected regret is defined as

$$\mathbb{E}[R_n] = nV^* - \sum_{t=1}^{n} \left\langle \hat{\pi}_t(\cdot), V(\cdot) \right\rangle. \tag{15}$$

Theorem 3 *Consider an E3W policy applied to the tree. Let define $\mathcal{D}_{\Omega^*}(x,y) = \Omega^*(x) - \Omega^*(y) - \nabla\Omega^*(y)(x-y)$ as the Bregman divergence between x and y, The expected pseudo regret R_n satisfies*

$$\begin{aligned} \mathbb{E}[R_n] \leq & -\tau\Omega(\hat{\pi}) + \sum_{t=1}^{n} \mathcal{D}_{\Omega^*}(\hat{V}_t(\cdot) + V(\cdot), \hat{V}_t(\cdot)) \\ & + \mathcal{O}(\frac{n}{\log n}). \end{aligned} \tag{16}$$

Table 1. List of entropy regularizers with Legendre-Fenchel transforms and maximizing arguments.

Entropy	**Regularizer** $\Omega(\pi_s)$	**Legendre-Fenchel** $\Omega^*(Q_s)$	**Max argument** $\nabla\Omega^*(Q_s)$
Maximum	$\sum_a \pi(a\|s)\log\pi(a\|s)$	$\tau\log\sum_a e^{\frac{Q(s,a)}{\tau}}$	$\frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_b e^{\frac{Q(s,b)}{\tau}}}$
Relative	$\mathrm{D_{KL}}(\pi_t(a\|s)\|\|\pi_{t-1}(a\|s))$	$\tau\log\sum_a \pi_{t-1}(a\|s)e^{\frac{Q_t(s,a)}{\tau}}$	$\frac{\pi_{t-1}(a\|s)e^{\frac{Q_t(s,a)}{\tau}}}{\sum_b \pi_{t-1}(b\|s)e^{\frac{Q_t(s,b)}{\tau}}}$
Tsallis	$\frac{1}{2}(\\| \pi(a\|s) \\|_2^2 -1)$	Equation (11)	Equation (12)

This theorem bounds the regret of E3W for a generic convex regularizer Ω; the regret bounds for each entropy regularizer can be easily derived from it. Let $m = \min_a \nabla\Omega^*(a|s)$.

Corollary 1 *Maximum entropy regret:*
$\mathbb{E}[R_n] \leq \tau(\log|\mathcal{A}|) + \frac{n|\mathcal{A}|}{\tau} + \mathcal{O}(\frac{n}{\log n})$.

Corollary 2 *Relative entropy regret:*
$\mathbb{E}[R_n] \leq \tau(\log|\mathcal{A}| - \frac{1}{m}) + \frac{n|\mathcal{A}|}{\tau} + \mathcal{O}(\frac{n}{\log n})$.

Corollary 3 *Tsallis entropy regret:*
$\mathbb{E}[R_n] \leq \tau(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}) + \frac{n|\mathcal{K}|}{2} + \mathcal{O}(\frac{n}{\log n})$.

Remarks. The regret bound of UCT and its variance have already been analyzed for non-regularized MCTS with binary tree (Coquelin & Munos, 2007). On the contrary, our regret bound analysis in Theorem 3 applies to generic regularized MCTS. From the specialized bounds in the corollaries, we observe that the maximum and relative entropy share similar results, although the bounds for relative entropy are slightly smaller due to $\frac{1}{m}$. Remarkably, the bounds for Tsallis entropy become tighter for increasing number of actions, which translates in limited regret in problems with high branching factor. This result establishes the advantage of Tsallis entropy in complex problems w.r.t. to other entropy regularizers, as empirically confirmed in Section 5.

4.2. Error analysis

We analyse the error of the regularized value estimate at the root node $n(s)$ w.r.t. the optimal value: $\varepsilon_\Omega = V_\Omega(s) - V^*(s)$.

Theorem 4 *For any* $\delta > 0$ *and generic convex regularizer* Ω*, with some constant* $C, \hat{C}$*, with probability at least* $1-\delta$*,* ε_Ω *satisfies*

$$-\sqrt{\frac{\hat{C}\sigma^2\log\frac{C}{\delta}}{2N(s)}} - \frac{\tau(U_\Omega - L_\Omega)}{1-\gamma} \leq \varepsilon_\Omega \leq \sqrt{\frac{\hat{C}\sigma^2\log\frac{C}{\delta}}{2N(s)}}. \tag{17}$$

To the best of our knowledge, this theorem provides the first result on the error analysis of value estimation at the root node of convex regularization in MCTS. To give a better understanding of the effect of each entropy regularizer in Table 1, we specialize the bound in Equation 17 to each of them. From (Lee et al., 2018), we know that for maximum entropy $\Omega(\pi_t) = \sum_a \pi_t \log\pi_t$, we have $-\log|\mathcal{A}| \leq \Omega(\pi_t) \leq 0$; for relative entropy $\Omega(\pi_t) = \mathrm{KL}(\pi_t||\pi_{t-1})$, if we define $m = \min_a \pi_{t-1}(a|s)$, then we can derive $0 \leq \Omega(\pi_t) \leq -\log|\mathcal{A}| + \log\frac{1}{m}$; and for Tsallis entropy $\Omega(\pi_t) = \frac{1}{2}(\| \pi_t \|_2^2 -1)$, we have $-\frac{|\mathcal{A}|-1}{2|\mathcal{A}|} \leq \Omega(\pi_t) \leq 0$. Then, defining $\Psi = \sqrt{\frac{\hat{C}\sigma^2\log\frac{C}{\delta}}{2N(s)}}$,

Corollary 4 *Maximum entropy error:*

$$-\Psi - \frac{\tau\log|\mathcal{A}|}{1-\gamma} \leq \varepsilon_\Omega \leq \Psi.$$

Corollary 5 *Relative entropy error:*

$$-\Psi - \frac{\tau(\log|\mathcal{A}| - \log\frac{1}{m})}{1-\gamma} \leq \varepsilon_\Omega \leq \Psi.$$

Corollary 6 *Tsallis entropy error:*

$$-\Psi - \frac{|\mathcal{A}|-1}{2|\mathcal{A}|}\frac{\tau}{1-\gamma} \leq \varepsilon_\Omega \leq \Psi.$$

These results show that when the number of actions $|\mathcal{A}|$ is large, TENTS enjoys the smallest error; moreover, we also see that lower bound of RENTS is always smaller than for MENTS.

5. Empirical evaluation

In this section, we empirically evaluate the benefit of the proposed entropy-based MCTS regularizers. First, we complement our theoretical analysis with an empirical study of the synthetic tree toy problem introduced in Xiao et al. (2019), which serves as a simple scenario to give an interpretable demonstration of the effects of our theoretical results in practice. Second, we compare to AlphaGo (Silver et al., 2016), recently introduced to enable MCTS to

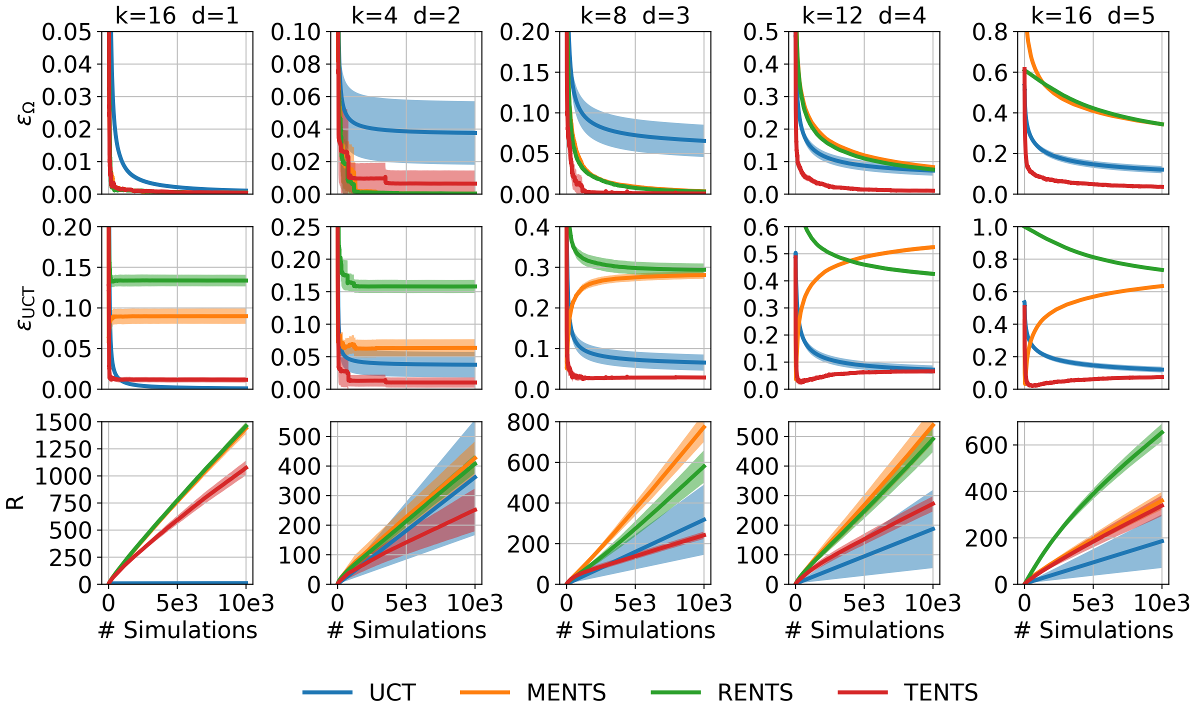


Figure 1. For each algorithm, we show the convergence of the value estimate at the root node to the respective optimal value (top), to the UCT optimal value (middle), and the regret (bottom).

solve large scale problems with high branching factor. Our implementation is a simplified version of the original algorithm, where we remove various tricks in favor of better interpretability. For the same reason, we do not compare with the most recent and state-of-the-art MuZero (Schrittwieser et al., 2019), as this is a slightly different solution highly tuned to maximize performance, and a detailed description of its implementation is not available.

The learning time of AlphaZero can be slow in problems with high branching factor, due to the need of a large number of MCTS simulations for obtaining good estimates of the randomly initialized action-values. To overcome this problem, AlphaGo (Silver et al., 2016) initializes the action-values using the values retrieved from a pretrained network, which is kept fixed during the training.

5.1. Synthetic tree

This toy problem is introduced in Xiao et al. (2019) to highlight the improvement of MENTS over UCT. It consists of a tree with branching factor k and depth d. Each edge of the tree is assigned a random value between 0 and 1. At each leaf, a Gaussian distribution is used as an evaluation function resembling the return of random rollouts. The mean of the Gaussian distribution is the sum of the values assigned to the edges connecting the root node to the leaf, while the standard deviation is $\sigma = 0.05$[2]. For stability, all the means are normalized between 0 and 1. As in Xiao et al. (2019), we create 5 trees on which we perform 5 different runs in each, resulting in 25 experiments, for all the combinations of branching factor $k = \{2, 4, 6, 8, 10, 12, 14, 16\}$ and depth $d = \{1, 2, 3, 4, 5\}$, computing: (i) the value estimation error at the root node w.r.t. the regularized optimal value: $\varepsilon_\Omega = V_\Omega - V^*_\Omega$; (ii) the value estimation error at the root node w.r.t. the unregularized optimal value: $\varepsilon_{\text{UCT}} = V_\Omega - V^*_{\text{UCT}}$; (iii) the regret R as in Equation (14). For a fair comparison, we use fixed $\tau = 0.1$ and $\epsilon = 0.1$ across all algorithms. Figure 1 and 2 show how UCT and each regularizer behave for different configurations of the tree. We observe that, while RENTS and MENTS converge slower for increasing tree sizes, TENTS is robust w.r.t. the size of the tree and almost always converges faster than all other methods to the respective optimal value. Notably, the optimal value of TENTS seems to be very close to the one of UCT, i.e. the optimal value of the unregularized objective, and also converges faster than the one estimated by UCT, while MENTS and RENTS are considerably further from this value. In terms of regret, UCT explores less than the regularized methods and it is less prone to high regret, at the cost of slower convergence time. Nevertheless, the regret of

[2] The value of the standard deviation is not provided in Xiao et al. (2019). After trying different values, we observed that our results match the one in Xiao et al. (2019) when using $\sigma = 0.05$.

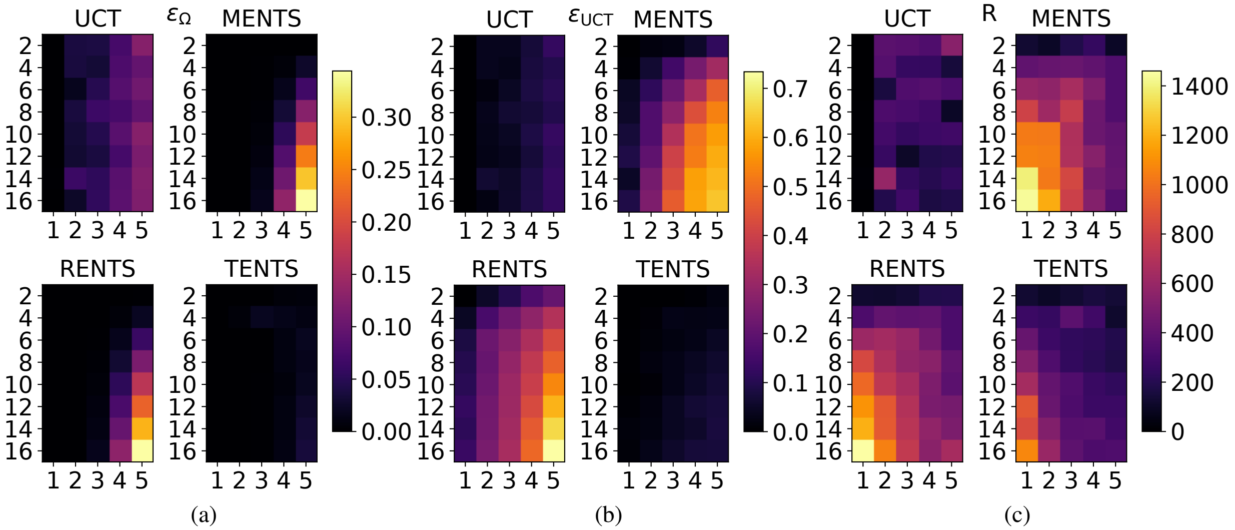
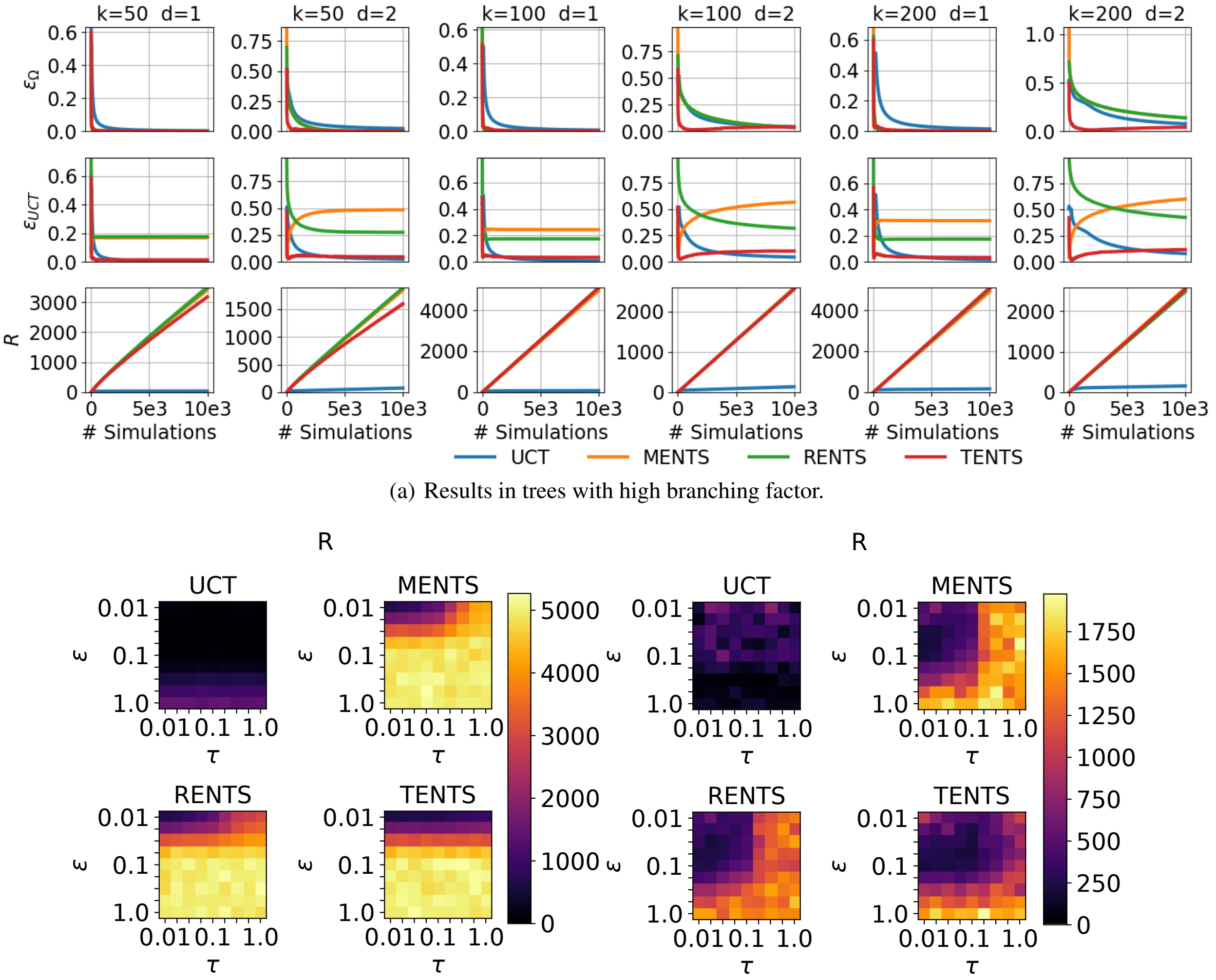


Figure 2. For different branching factor k (rows) and depth d (columns), the heatmaps show: the absolute error of the value estimate at the root node after the last simulation of each algorithm w.r.t. the respective optimal value (a), and w.r.t. the optimal value of UCT (b); regret at the root node (c).



(a) Results in trees with high branching factor.

(b) $k = 100$, $d = 1$.

(c) $k = 8$, $d = 3$.

Figure 3. High branching factor trees (a), regret sensitivity study w.r.t. ε and τ (b, c).

TENTS is the smallest between the ones of the other regularizers, which seem to explore too much. In Figure 3(a), we show further results evincing the advantages of TENTS over the baselines in problems with high branching factor, in terms of approximation error and regret. Finally, in Figures 3(b) and 3(c) we carry out a sensitivity analysis of each algorithm w.r.t. the values of the exploration coefficient ε and τ in two different trees. Note that ε is only used by E3W to choose whether to sample uniformly or from the regularized policy. We observe that the choice of τ does not significantly impact the regret of TENTS, as opposed to the other methods. These results show a general superiority of TENTS in this toy problem, also confirming our theoretical findings about the advantage of TENTS in terms of approximation error (Corollary 6) and regret (Corollary 3), in problems with many actions.

5.2. Entropy-regularized AlphaGo

Atari. Atari 2600 (Bellemare et al., 2013) is a popular benchmark for testing deep RL methodologies (Mnih et al., 2015; Van Hasselt et al., 2016; Bellemare et al., 2017) but still relatively disregarded in MCTS. We use a Deep Q-Network, pretrained using the same experimental setting of Mnih et al. (2015), to initialize the action-value function of each node after expansion as $Q_{init}(s,a) = (Q(s,a) - V(s))/\tau$, for MENTS and TENTS, as done in Xiao et al. (2019). For RENTS we init $Q_{init}(s,a) = \log P_{\text{prior}}(a|s)) + (Q(s,a) - V(s))/\tau$, where P_{prior} is the Boltzmann distribution induced by action-values $Q(s,.)$ computed from the network. Each experimental run consists of 512 MCTS simulations. The temperature τ is optimized for each algorithm and game via grid-search between 0.01 and 1. The discount factor is $\gamma = 0.99$, and for PUCT the exploration constant is $c = 0.1$. Table 2 shows the performance, in terms of cumulative reward, of standard AlphaGo with PUCT and our three regularized versions, on 22 Atari games. Moreover, we test also AlphaGo using the MaxMCTS backup (Khandelwal et al., 2016) for further comparison with classic baselines. We observe that regularized MCTS dominates other baselines, in particular TENTS achieves the highest scores in all the 22 games, showing that sparse policies are more effective in Atari. In particular, TENTS significantly outperforms the other methods in the games with many actions, e.g. Asteroids, Phoenix, confirming the results obtained in the synthetic tree experiment, explained by corollaries 3 and 6 on the benefit of TENTS in problems with high-branching factor.

6. Related Work

Entropy regularization is a common tool for controlling exploration in Reinforcement Learning (RL) and has lead to several successful methods (Schulman et al., 2015; Haarnoja et al., 2018; Schulman et al., 2017a; Mnih et al., 2016). Typically specific forms of entropy are utilized such as maximum entropy (Haarnoja et al., 2018) or relative entropy (Schulman et al., 2015). This approach is an instance of the more generic duality framework, commonly used in convex optimization theory. Duality has been extensively studied in game theory (Shalev-Shwartz & Singer, 2006; Pavel, 2007) and more recently in RL, for instance considering mirror descent optimization (Montgomery & Levine, 2016; Mei et al., 2019), drawing the connection between MCTS and regularized policy optimization (Grill et al., 2020), or formalizing the RL objective via Legendre-Rockafellar duality (Nachum & Dai, 2020a). Recently (Geist et al., 2019) introduced regularized Markov Decision Processes, formalizing the RL objective with a generalized form of convex regularization, based on the Legendre-Fenchel transform. In this paper, we provide a novel study of convex regularization in MCTS, and derive relative entropy (KL-divergence) and Tsallis entropy regularized MCTS algorithms, i.e. RENTS and TENTS respectively. Note that the recent maximum entropy MCTS algorithm MENTS (Xiao et al., 2019) is a special case of our generalized regularized MCTS. Unlike MENTS, RENTS can take advantage of any action distribution prior, in the experiments the prior is derived using Deep Q-learning (Mnih et al., 2015). On the other hand, TENTS allows for sparse action exploration and thus higher dimensional action spaces compared to MENTS. Several works focus on modifying classical MCTS to improve exploration. UCB1-tuned (Auer et al., 2002) modifies the upper confidence bound of UCB1 to account for variance in order to improve exploration. (Tesauro et al., 2012) proposes a Bayesian version of UCT, which obtains better estimates of node values and uncertainties given limited experience. Many heuristic approaches based on specific domain knowledge have been proposed, such as adding a bonus term to value estimates (Gelly & Wang, 2006; Teytaud & Teytaud, 2010; Childs et al., 2008; Kozelek, 2009; Chaslot et al., 2008) or prior knowledge collected during policy search (Gelly & Silver, 2007; Helmbold & Parker-Wood, 2009; Lorentz, 2010; Tom, 2010; Hoock et al., 2010). (Khandelwal et al., 2016) formalizes and analyzes different on-policy and off-policy complex backup approaches for MCTS planning based on RL techniques. (Vodopivec et al., 2017) proposes an approach called SARSA-UCT, which performs the dynamic programming backups using SARSA (Rummery, 1995). Both (Khandelwal et al., 2016) and (Vodopivec et al., 2017) directly borrow value backup ideas from RL to estimate the value at each tree node, but they do not provide any proof of convergence.

7. Conclusion

We introduced a theory of convex regularization in Monte-Carlo Tree Search (MCTS) based on the Legendre-Fenchel

Table 2. Average score in Atari over 100 seeds per game. Bold denotes no statistically significant difference to the highest mean (t-test, $p < 0.05$). Bottom row shows # no difference to highest mean.

	UCT	MaxMCTS	MENTS	RENTS	TENTS
Alien	**1,486.80**	**1,461.10**	**1,508.60**	**1,547.80**	**1,568.60**
Amidar	115.62	**124.92**	**123.30**	**125.58**	**121.84**
Asterix	4,855.00	**5,484.50**	**5,576.00**	**5,743.50**	**5,647.00**
Asteroids	873.40	899.60	1,414.70	1,486.40	**1,642.10**
Atlantis	35,182.00	**35,720.00**	**36,277.00**	35,314.00	**35,756.00**
BankHeist	475.50	458.60	**622.30**	**636.70**	**631.40**
BeamRider	**2,616.72**	**2,661.30**	**2,822.18**	2,558.94	**2,804.88**
Breakout	**303.04**	296.14	**309.03**	300.35	**316.68**
Centipede	1,782.18	1,728.69	**2,012.86**	**2,253.42**	**2,258.89**
DemonAttack	579.90	640.80	**1,044.50**	**1,124.70**	**1,113.30**
Enduro	**129.28**	124.20	128.79	**134.88**	**132.05**
Frostbite	1,244.00	1,332.10	**2,388.20**	**2,369.80**	**2,260.60**
Gopher	3,348.40	3,303.00	**3,536.40**	**3,372.80**	**3,447.80**
Hero	3,009.95	3,010.55	**3,044.55**	**3,077.20**	**3,074.00**
MsPacman	1,940.20	1,907.10	2,018.30	**2,190.30**	**2,094.40**
Phoenix	2,747.30	2,626.60	3,098.30	2,582.30	**3,975.30**
Qbert	7,987.25	8,033.50	8,051.25	8,254.00	**8,437.75**
Robotank	**11.43**	11.00	**11.59**	**11.51**	**11.47**
Seaquest	**3,276.40**	**3,217.20**	**3,312.40**	**3,345.20**	**3,324.40**
Solaris	895.00	923.20	**1,118.20**	**1,115.00**	**1,127.60**
SpaceInvaders	778.45	**835.90**	**832.55**	**867.35**	**822.95**
WizardOfWor	685.00	666.00	**1,211.00**	**1,241.00**	**1,231.00**
# Highest mean	6/22	7/22	17/22	16/22	**22/22**

transform. We proved that a generic strongly convex regularizer has an exponential convergence rate for the selection of the optimal action at the root node. Our result gives theoretical motivations to previous results specific to maximum entropy regularization. Furthermore, we provided the first study of the regret of MCTS when using a generic strongly convex regularizer, and an analysis of the error between the regularized value estimate at the root node and the optimal regularized value. We use these results to motivate the use of entropy regularization in MCTS, considering maximum, relative, and Tsallis entropy, and we specialized our regret and approximation error bounds to each entropy-regularizer. We tested our regularized MCTS algorithm in a simple toy problem, where we give an empirical evidence of the effect of our theoretical bounds for the regret and approximation error. Finally, we introduced the use of convex regularization in AlphaGo, and carried out experiments on several Atari games. Overall, our empirical results show the advantages of convex regularization, and in particular the superiority of Tsallis entropy w.r.t. other entropy-regularizers.

Future developments of this work can investigate the possibility of mixing UCT and the regularized policy in a complementary manner. In our empirical results, we observed that UCT enjoys a better cumulative regret than regularized policies, while regularized policies have a better one-step regret due to the exponential convergence rate. Considering that cumulative regret and one-step regret can be both significant objectives to minimize according to the problem at hand (Bubeck et al., 2009), studying theoretically sound ways of mixing UCT and regularized policy seems a promising idea.

Acknowledgments

This project has received funding from the German Research Foundation project PA 3179/1-1 (ROBOLEAP).

References

Abernethy, J., Lee, C., and Tewari, A. Fighting bandits with a new kind of smoothness. *arXiv preprint arXiv:1512.04152*, 2015.

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

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 449–458. JMLR. org, 2017.

Bellman, R. The theory of dynamic programming. Technical report, Rand corp santa monica ca, 1954.

Belousov, B. and Peters, J. Entropic regularization of markov decision processes. *Entropy*, 21(7):674, 2019.

Bubeck, S., Munos, R., and Stoltz, G. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pp. 23–37. Springer, 2009.

Buesing, L., Heess, N., and Weber, T. Approximate inference in discrete distributions with monte carlo tree search and value functions. In *International Conference on Artificial Intelligence and Statistics*, pp. 624–634. PMLR, 2020.

Chaslot, G., Winands, M., Herik, J. V. D., Uiterwijk, J., and Bouzy, B. Progressive strategies for monte-carlo tree search. *New Mathematics and Natural Computation*, 4 (03):343–357, 2008.

Childs, B. E., Brodeur, J. H., and Kocsis, L. Transpositions and move groups in monte carlo tree search. In *2008 IEEE Symposium On Computational Intelligence and Games*. IEEE, 2008.

Coquelin, P.-A. and Munos, R. Bandit algorithms for tree search. *arXiv preprint cs/0703062*, 2007.

Coulom, R. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pp. 72–83. Springer, 2006.

Geist, M. and Scherrer, B. L1-penalized projected bellman residual. In *Proceedings of the European Workshop on Reinforcement Learning (EWRL 2011)*, Lecture Notes in Computer Science (LNCS). Springer Verlag - Heidelberg Berlin, september 2011.

Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pp. 2160–2169, 2019.

Gelly, S. and Silver, D. Combining online and offline knowledge in uct. In *Proceedings of the 24th international conference on Machine learning*, pp. 273–280. ACM, 2007.

Gelly, S. and Wang, Y. Exploration exploitation in go: Uct for monte-carlo go. In *NIPS: Neural Information Processing Systems Conference On-line trading of Exploration and Exploitation Workshop*, 2006.

Grill, J.-B., Altché, F., Tang, Y., Hubert, T., Valko, M., Antonoglou, I., and Munos, R. Monte-carlo tree search as regularized policy optimization. *arXiv preprint arXiv:2007.12509*, 2020.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

Helmbold, D. P. and Parker-Wood, A. All-moves-as-first heuristics in monte-carlo go. In *IC-AI*, pp. 605–610, 2009.

Hoock, J.-B., Lee, C.-S., Rimmel, A., Teytaud, F., Wang, M.-H., and Teytaud, O. Intelligent agents for the game of go. *IEEE Computational Intelligence Magazine*, 2010.

Khandelwal, P., Liebman, E., Niekum, S., and Stone, P. On the analysis of complex backup strategies in monte carlo tree search. In *International Conference on Machine Learning*, 2016.

Kocsis, L., Szepesvári, C., and Willemson, J. Improved monte-carlo search, 2006.

Kozelek, T. Methods of mcts and the game arimaa, 2009.

Lee, K., Choi, S., and Oh, S. Sparse markov decision processes with causal sparse tsallis entropy regularization for reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):1466–1473, 2018.

Lorentz, R. J. Improving monte–carlo tree search in havannah. In *International Conference on Computers and Games*, pp. 105–115. Springer, 2010.

Mei, J., Xiao, C., Huang, R., Schuurmans, D., and Müller, M. On principled entropy exploration in policy optimization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 3130–3136. AAAI Press, 2019.

Mensch, A. and Blondel, M. Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, pp. 3462–3471, 2018.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Montgomery, W. H. and Levine, S. Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems*, pp. 4008–4016, 2016.

Nachum, O. and Dai, B. Reinforcement learning via fenchel-rockafellar duality. *CoRR*, abs/2001.01866, 2020a.

Nachum, O. and Dai, B. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020b.

Niculae, V. and Blondel, M. A regularized framework for sparse and structured neural attention. *arXiv preprint arXiv:1705.07704*, 2017.

Pavel, L. An extension of duality to a game-theoretic framework. *Automatica*, 43(2):226 – 237, 2007.

Rummery, G. A. *Problem solving with reinforcement learning*. PhD thesis, University of Cambridge Ph. D. dissertation, 1995.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., and Silver, D. Mastering atari, go, chess and shogi by planning with a learned model, 2019.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, pp. 1889–1897, 2015.

Schulman, J., Chen, X., and Abbeel, P. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

Shalev-Shwartz, S. and Singer, Y. Convex repeated games and fenchel duality. *Advances in neural information processing systems*, 19:1265–1272, 2006.

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. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017a.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017b.

Sutton, R. S. and Barto, A. G. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

Tesauro, G., Rajan, V., and Segal, R. Bayesian inference in monte-carlo tree search. *arXiv preprint arXiv:1203.3519*, 2012.

Teytaud, F. and Teytaud, O. On the huge benefit of decisive moves in monte-carlo tree search algorithms. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pp. 359–364. IEEE, 2010.

Tom, D. Investigating uct and rave: Steps towards a more robust method, 2010.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Vodopivec, T., Samothrakis, S., and Ster, B. On monte carlo tree search and reinforcement learning. *Journal of Artificial Intelligence Research*, 60:881–936, 2017.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Xiao, C., Huang, R., Mei, J., Schuurmans, D., and Müller, M. Maximum entropy monte-carlo planning. In *Advances in Neural Information Processing Systems*, pp. 9516–9524, 2019.

Yee, T., Lisỳ, V., Bowling, M. H., and Kambhampati, S. Monte carlo tree search in continuous action spaces with execution uncertainty. In *IJCAI*, pp. 690–697, 2016.

Zimmert, J. and Seldin, Y. An optimal algorithm for stochastic and adversarial bandits. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 467–475. PMLR, 2019.