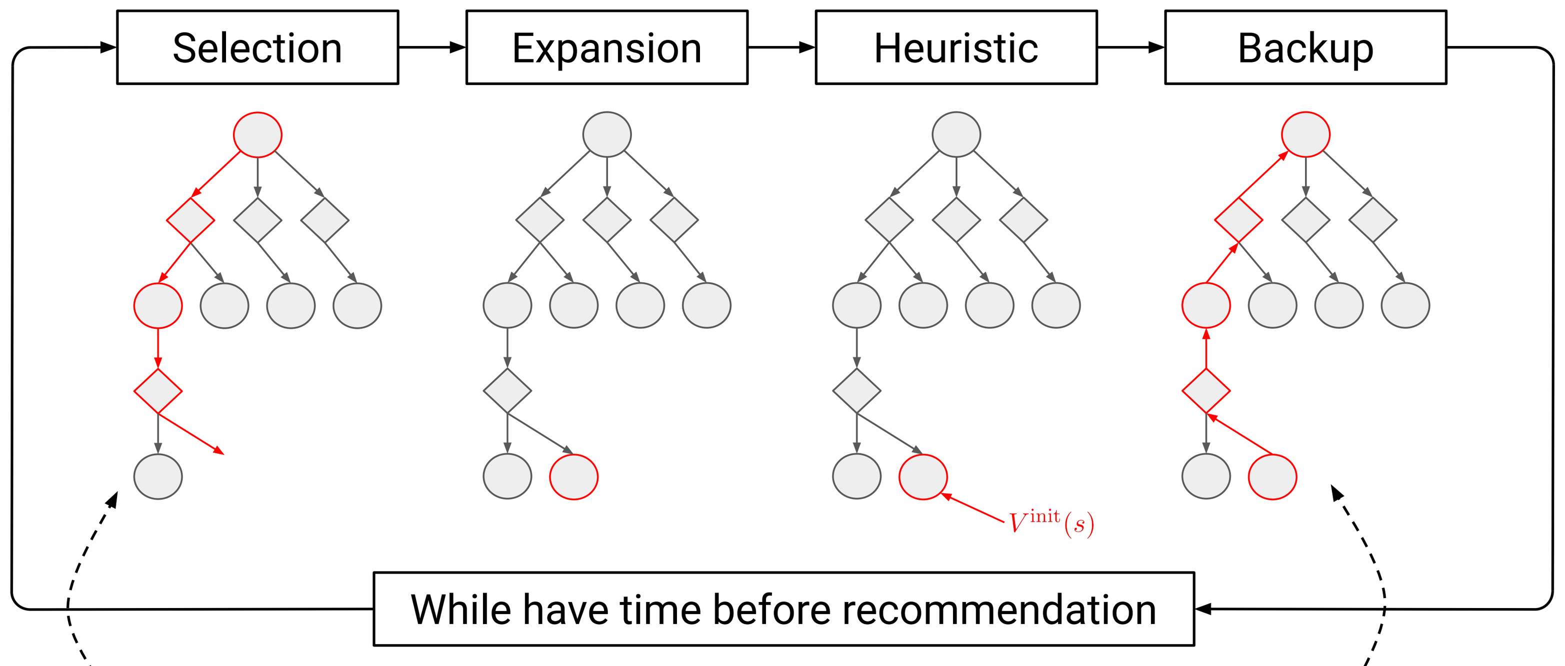
Monte Carlo Tree Search with Boltzmann Exploration

Michael Painter, Mohamed Baioumy, Nick Hawes, Bruno Lacerda
Oxford Robotics Institute, University of Oxford
[mpainter, mohamed, nickh, bruno]@robots.ox.ac.uk

Boltzmann Tree Search (BTS)

- BTS follows the Monte Carlo Tree Search (MCTS) schema:



$$\pi_{\text{BTS}}(a|s) = (1-\lambda_s)\rho_{\text{BTS}}(a|s) + \frac{\lambda_s}{|\mathcal{A}|}$$
$$\rho_{\text{BTS}}(a|s) \propto \exp\left(\frac{1}{\alpha}\left(\hat{Q}(s,a)\right)\right)$$

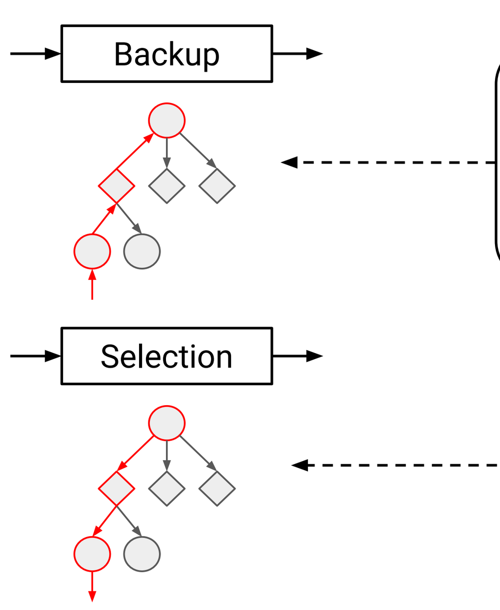
$$\hat{Q}(s_t,a_t) \leftarrow R(s_t,a_t) + \sum_{s'\in\text{Succ}(s_t,a_t)} \left(\frac{N(s')}{N(s_t,a_t)}\hat{V}(s')\right)$$
$$\hat{V}(s_t) \leftarrow \max_{a\in\mathcal{A}} \hat{Q}(s_t,a)$$

- Actions are sampled from a Boltzmann distribution
- Value estimates updated with Bellman backups

$s_0, a_0, s_1, a_1, s_2, a_2, \ldots$ - States and actions sampled in selection phase
V^{init} - Heuristic function, used to initialise value estimates (can be stochastic, such as the return from a rollout)
$R(s,a)$ - Reward for taking action a from state s
$N(s)$ - Number of visits to decision node associated with state s
$N(s,a)$ - Number of visits to chance node associated with taking action a from state
α - Search temperature parameter
$\lambda_s = \min(1, \epsilon/\log(e+N(s)))$ - Exploration parameter
$\text{Succ}(s,a)$ - The set of successor states from taking action a in state s

Decaying ENtropy Tree Search (DENTS)

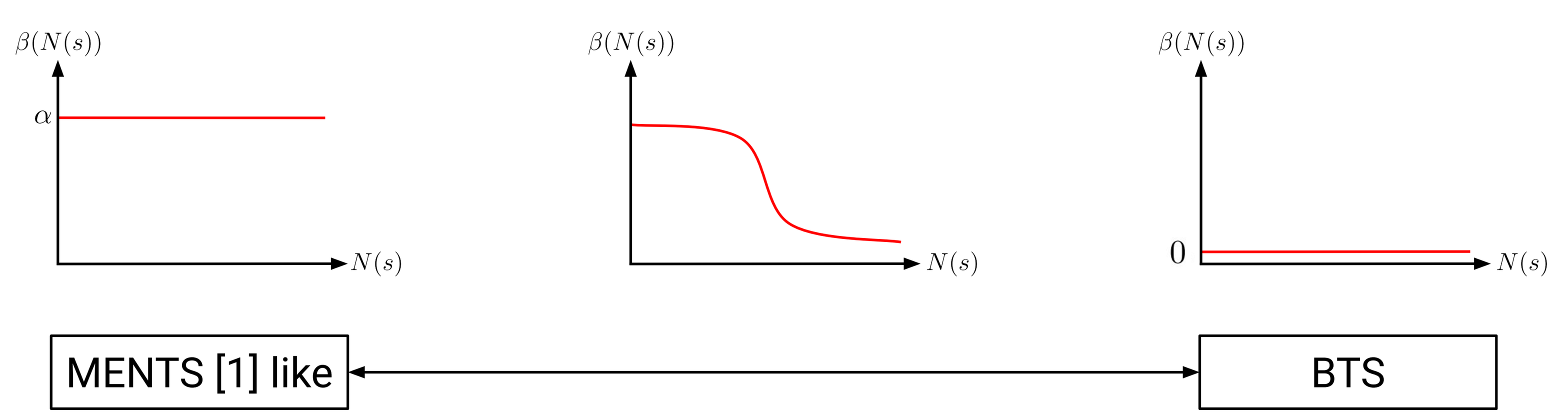
- Builds on top of Boltzmann Tree Search
- Computes entropy over subtrees in backups
- Uses entropy values as an exploration term in search policy



$$\mathcal{H}_V(s_t) \leftarrow \mathcal{H}(\pi_{\text{DENTS}}(\cdot|s_t)) + \sum_{a\in\mathcal{A}} \pi_{\text{DENTS}}(a|s_t)\mathcal{H}_Q(s_t,a)$$
$$\mathcal{H}_Q(s_t,a_t) \leftarrow \sum_{s'\in\text{Succ}(s_t,a_t)} \frac{N(s')}{N(s_t,a_t)}\mathcal{H}_V(s')$$

$$\pi_{\text{DENTS}}(a|s) = (1-\lambda_s)\rho_{\text{DENTS}}(a|s) + \frac{\lambda_s}{|\mathcal{A}|}$$
$$\rho_{\text{DENTS}}(a|s) \propto \exp\left(\frac{1}{\alpha}\left(\hat{Q}(s,a) + \beta(N(s))\mathcal{H}_Q(s,a)\right)\right)$$

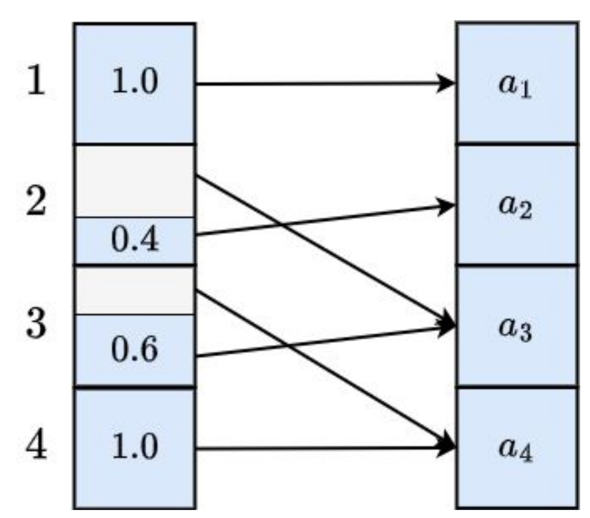
- Entropy weighted by function with respect to #visits to node
- Different functions give a range of search behaviours



$\mathcal{H}_V(s)$ - Shannon entropy of search policy in subtree rooted at decision node associated with state s
$\mathcal{H}_Q(s,a)$ - Shannon entropy of search policy in subtree rooted at chance node associated with taking action a from state s
β - Entropy weight function parameter, maps the #visits at a node to the weighting to use for the entropy term in the search policy at that node

Alias Method [2,3]

- Given a categorical distribution, an alias table can be built in linear time and sampled from in constant time
- Example:
  - Sample an integer between 1 and 4
  - Sample a uniform random number between 0.0 and 1.0 in [0,1]



$\Pr(a_1) = 0.25$
$\Pr(a_2) = 0.1$
$\Pr(a_3) = 0.3$
$\Pr(a_4) = 0.35$

- Amortised O(1) action sampling in MCTS with stochastic search policy:
  - If ($N(s)$ mod $|\mathcal{A}|$) == 0 then recompute alias table
  - Sample from alias table
- Comes with a cost of not using most up to date policy

Theoretical Results

- Analysis of algorithms using simple regret [4]:

$$\text{reg}(s,\psi) = V^*(s) - V^{\psi}(s)$$

- BTS and DENTS recommendations use Q-value estimates:

$$\psi_{\text{BTS}}(s) = \psi_{\text{DENTS}}(s) = \text{argmax}_{a\in\mathcal{A}}\hat{Q}(s,a)$$

- Expected simple regret of BTS and DENTS tends to zero with an exponential concentration bound:

Theorem 4.1. *For any MDP $\mathcal{M}$, after running n trials of the BTS algorithm with a root node of s_0, there exists constants $C, k > 0$ such that for all $\varepsilon > 0$ we have $\mathbb{E}[\text{reg}(s_0,\psi_{\text{BTS}})] \leq C\exp(-kn)$, and also $\hat{V}(s_0) \xrightarrow{p} V^*(s_0)$ as $n \to \infty$.*

Theorem 4.2. *For any MDP $\mathcal{M}$, after running n trials of the DENTS algorithm with a root node of s_0, if β is a bounded function, then there exists constants $C, k > 0$ such that for all $\varepsilon > 0$ we have $\mathbb{E}[\text{reg}(s_0,\psi_{\text{DENTS}})] \leq C\exp(-kn)$, and also $\hat{V}(s_0) \xrightarrow{p} V^*(s_0)$ as $n \to \infty$.*

- BTS and DENTS can use average returns and still converge if the search temperature is decayed:

Proposition B.1. *For any $\alpha_{\text{fix}} > 0$, there is an MDP $\mathcal{M}$ such that AR-BTS with $\alpha(m) = \alpha_{\text{fix}}$ is not consistent: $\mathbb{E}[\text{reg}(s_0,\psi^n_{\text{AR-BTS}})] \not\to 0$ as $n \to \infty$.*

Theorem B.2. *For any MDP $\mathcal{M}$, if $\alpha(m) \to 0$ as $m \to \infty$ then $\mathbb{E}[\text{reg}(s_0,\psi^n_{\text{AR-BTS}})] \to 0$ as $n \to \infty$, where n is the number of trials.*

Theorem B.3. *For any MDP $\mathcal{M}$, if $\alpha(m) \to 0$ and $\beta(m) \to 0$ as $m \to \infty$ then $\mathbb{E}[\text{reg}(s_0,\psi^n_{\text{AR-DENTS}})] \to 0$ as $n \to \infty$, where n is the number of trials.*

Links

THTS++ github:

Comparison

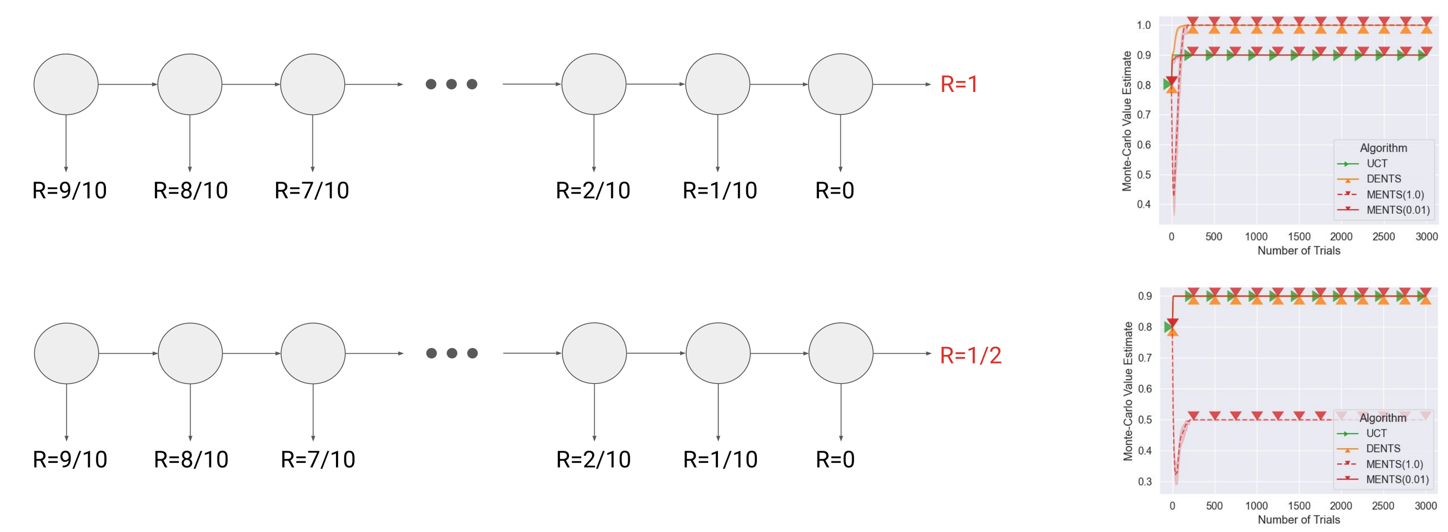
- Overview of differences between some MCTS algorithms:

	UCT [5]	MENTS [1]	BTS	DENTS
Consistent for any setting of parameters	✓	x	✓	✓
Actions sampled stochastically (i.e. can use Alias method)	x	✓	✓	✓
Utilises entropy for exploration	x	✓	x	✓
Optimises for cumulative regret	✓	x	x	x
Optimises for simple regret	x	x	✓	✓

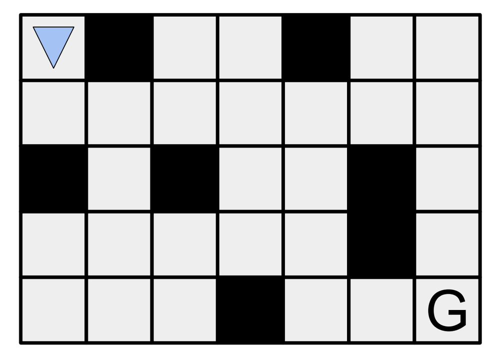
- Consistency refers to the recommended action/policy converging to the optimal action/policy in the limit
  - I.e. running more trials should improve recommendations

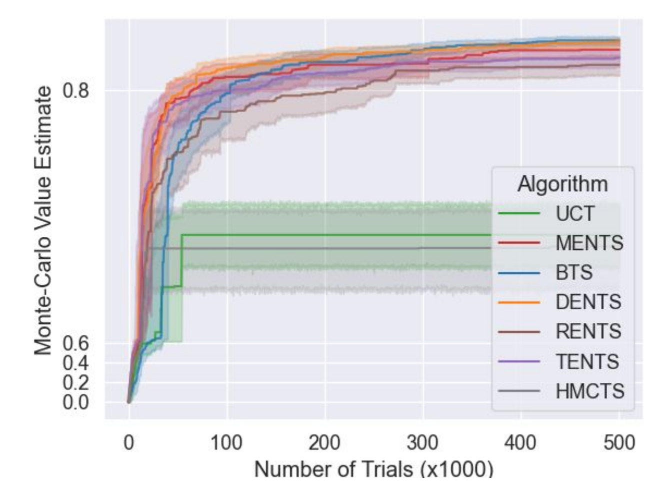
Empirical Results

- Minimal motivating example where the maximum entropy objective (see MENTS) can lead to unwanted behaviour:



- Frozen Lake (reward of 0.99^T for reaching goal after T steps):





- Go (5s planning time per move):

Black\White	PUCT	AR-BTS	AR-DENTS	#Trials/move
PUCT		17-33	15-35	1054
AR-BTS	25-25 (58-42)		15-35	5375
AR-DENTS	23-27 (58-42)	15-35 (50-50)		4677

References

[1] Chenjun Xiao, Ruitong Huang, Jincheng Mei, Dale Schuurmans, and Martin Müller. Maximum entropy monte-carlo planning. Advances in Neural Information Processing Systems, 32, 2019.
[2] Alastair J Walker. New fast method for generating discrete random numbers with arbitrary frequency distributions. Electronics Letters, 8(10):127–128, 1974.
[3] Michael D Vose. A linear algorithm for generating random numbers with a given distribution. IEEE Transactions on software engineering, 17(9):972–975, 1991.
[4] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In International conference on Algorithmic learning theory, pages 23–37. Springer, 2009.
[5] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In European conference on machine learning, pages 282–293. Springer, 2006.
[6] Tuan Q Dam, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. Convex regularization in monte-carlo tree search. In International Conference on Machine Learning, pages 2365–2375. PMLR, 2021.
[7] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In International Conference on Machine Learning, pages 1238–1246. PMLR, 2013.