On Effective Parallelization of Monte Carlo Tree Search

Anji Liu[1], Yitao Liang[1], Ji Liu[2], Guy Van den Broeck[1], Jianshu Chen[3]
[1]Computer Science Department, UCLA
[2]Seattle AI Lab, Kwai Inc.
[3]Tencent AI Lab

Motivation: Monte Carlo Tree Search

MCTS is considered as one of the core methods in model-based reinforcement learning.

MCTS is slow, so it needs parallelization.

Go

figure credit: https://deepmind.com/research/case-studies/alphago-the-story-so-far

Chess

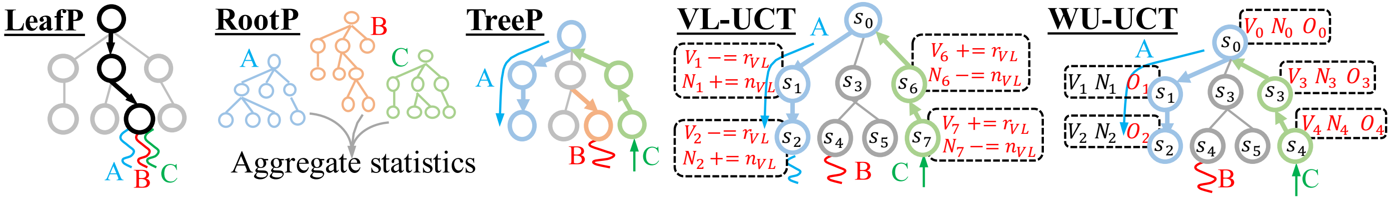
figure credit: https://www.businessinsider.com/chess-grandmaster-gary-kasparov-ai-artificial-intelligence-destroy-jobs-prediction-2020-2

Video games

figure credit: https://gym.openai.com/

Motivation: MCTS parallelization

Existing parallel MCTS algorithms:



However, it is unclear what are the pros and cons of existing algorithms and how to design effective parallel MCTS algorithms.

We seek to lay the first theoretical foundation for effective MCTS parallelization.

What is effective parallel MCTS?

We study the **performance loss** of parallel MCTS algorithms under a fixed **speedup** requirement.

Speedup

$$\text{speedup} = \frac{\text{runtime of the sequential MCTS}}{\text{runtime of algorithm } \mathbb{A} \text{ using } M \text{ workers}}$$

Performance loss: *excess regret*

The ***excess regret*** is defined as the difference between the **cumulative regret** of a parallel MCTS algorithm $\mathbb{A}$ and its sequential counterpart $\mathbb{A}_{seq}$ (i.e., $Regret_{\mathbb{A}}(n) - Regret_{\mathbb{A}_{seq}}(n)$):

$$\text{Regret}_{\mathbb{A}}(n) := \sum_{i=1}^{n} \mathbb{E}\left[V_i^*(s_0) - V_i(s_0)\right]$$

s_0 - the root state

n - the number of rollouts

$V_i(s_0)$ - the value estimate of s_0 obtained in the i-th rollout of $\mathbb{A}$

$V_i^*(s_0)$ - the value estimate of s_0 obtained by an oracle algorithm

When will excess regret vanish?

The tree policy of UCT for selecting child nodes

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \Bigg\{ \underbrace{\boxed{\overline{Q}(s_t, a)}}_{\text{action value}} + c\sqrt{2 \ln \underbrace{\boxed{\frac{\sum_{a'} \overline{N}(s_t, a')}{\overline{N}(s_t, a)}}}_{\text{visit count}}} \Bigg\}$$

Two necessary conditions for achieving **vanishing excess regret**:

- Q: the action value gap $\bar{G}$ should be zero:

$$\overline{G}(s,a) := \Big| \underbrace{\mathbb{E}\big[\overline{Q}(s,a)\big]}_{\text{expected action value computed by the parallel algorithm } \mathbb{A}} - \underbrace{\mathbb{E}\big[Q_m^{\mathbb{A}_{\text{seq}}}(s,a)\big]}_{\text{expected action value computed by a virtual sequential algorithm } \mathbb{A}_{seq}} \Big|$$

- N: the algorithm should modify visit count using the number of incomplete simulations:

$$\overline{N}(s,a) \geq \underbrace{N(s,a)}_{\text{\# complete simulations}} + \underbrace{O(s,a)}_{\text{\# incomplete simulations}}$$

When will excess regret vanish?

The tree policy of UCT for selecting child nodes

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \left\{ \boxed{\overline{Q}(s_t, a)} + c \sqrt{2 \ln \boxed{\frac{\sum_{a'} \overline{N}(s_t, a')}{\overline{N}(s_t, a)}}} \right\}$$

action value

visit count

When the search tree's maximum depth is 2, WU-UCT [1] satisfies both necessary conditions. Furthermore, in this case WU-UCT theoretically enjoys vanishing excess regret.

Theorem 2. *Consider a tree search task $\mathbb{T}$ with maximum depth $D=2$ (abbreviate as the depth-2 tree search task): it contains a root node s and K feasible actions $\{a_i\}_{i=1}^K$ at s, which lead to terminal states $\{s_i\}_{i=1}^K$, respectively. Let $\mu_i := \mathbb{E}[V(s_i)]$, $\mu^* := \max_i \mu_i$ and $\Delta_k := \mu^* - \mu_k$, and further assume: $\forall i, V(s_i) - \mu_i$ is 1-subgaussian (Buldygin & Kozachenko, 1980). The cumulative regret of running WU-UCT (Liu et al., 2020) with n rollouts on $\mathbb{T}$ is upper bounded by:*

$$\underbrace{\sum_{k:\mu_k<\mu^*} \left(\frac{8}{\Delta_k} + 2\Delta_k\right) \ln n + \Delta_k}_{R_{\mathrm{UCT}}(n)} + \underbrace{4M \sum_{k:\mu_k<\mu^*} \frac{\Delta_k^2}{\sqrt{\ln n}}}_{\text{excess regret}},$$

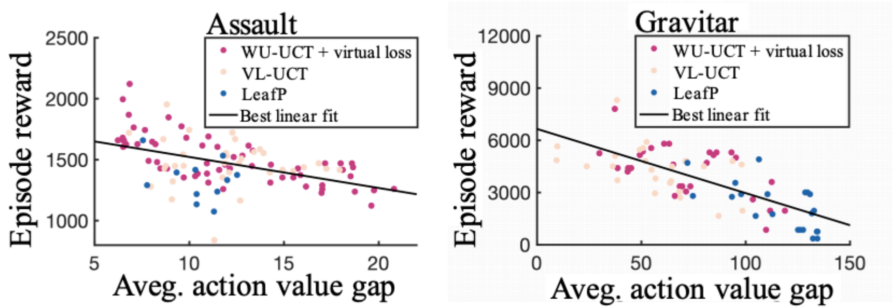
where $R_{\mathrm{UCT}}(n)$ is the cumulative regret of running the (sequential) UCT for n steps on $\mathbb{T}$.

Theory in practice: motivation

The action value gap $\bar{G}$

$$\overline{G}(s,a) := \left|\mathbb{E}\left[\overline{Q}(s,a)\right] - \mathbb{E}\left[Q_m^{\mathbb{A}_{\text{seq}}}(s,a)\right]\right|$$

The action value gap has **strong negative correlation** with the algorithm’s performance



Seek to design better parallel MCTS algorithms by minimizing the action value gap

Theory in practice: empirical evaluation

Environment	BU-UCT (ours)		WU-UCT	VL-UCT	LeafP	RootP
Alien	5320±231	†‡	**5938**±1839	4200±1086	4280±1016	5206±282
Boxing	**100**±0	†‡§	**100**±0	99±0	95±4	98±1
Breakout	**425**±30	‡§	408±21	390±33	331±45	281±27
Centipede	**1610419**±338295	†‡§	1163034±403910	439433±207601	162333±69575	184265±104405
Freeway	**32**±0		**32**±0	**32**±0	31±1	**32**±0
Gravitar	**5130**±499	‡	5060±568	4880±1162	3385±155	4160±1811
MsPacman	17279±6136	‡§	**19804**±2232	14000±2807	5378±685	7156±583
NameThisGame	**47066**±5911	*†‡§	29991±1608	23326±2585	25390±3659	27440±9533
RoadRunner	44920±1478	†‡§	**46720**±1359	24680±3316	25452±2977	38300±1191
Robotank	**121**±18	†‡§	101±19	86±13	80±11	78±13
Qbert	**15995**±2635	§	13992±5596	14620±5738	11655±5373	9465±3196
SpaceInvaders	**3428**±525	§	3393±292	2651±828	2435±1159	2543±809
Tennis	3±1	†‡§	**4**±1	−1±0	−1±0	0±1
TimePilot	**111100**±58919	*†‡§	55130±12474	32600±2165	38075±2307	45100±7421
Zaxxon	**42500**±4725	‡§	39085±6838	39579±3942	12300±821	13380±769

BU-UCT outperforms all baselines in 11 out of 15 Atari games.

Thank You

[1] Anji Liu, Jianshu Chen, Mingze Yu, Yu Zhai, Xuewen Zhou, and Ji Liu. Watch the unobserved: A simple approach to parallelizing monte carlo tree search. In *International Conference on Learning Representations*, April 2020. URL https://openreview.net/forum?id=BJlQtJSKDB.