Monte Carlo Tree Search

tree ← root

Until out of time

traverse tree root ⇝ leaf

tree policy

UCB (choosing a child w/o visits when one exists)

$\bar{r}_i + \sqrt{\frac{2 \ln T}{n_i}}$

exploit — exploration

reward for P1

expand if leaf expandable, add its children

non-terminal and non-zero visits

P2 max: $(c - \bar{r}_i) + \sqrt{\frac{2 \ln T}{n_i}}$

min $\bar{r}_i - \sqrt{\frac{2 \ln T}{n_i}}$

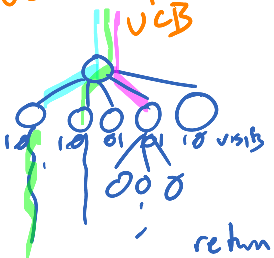
simulate play to terminal pos (from arbi. child if newly expanded)

can be random default policy

update backprop result up tree from start of simulation ⇝ root

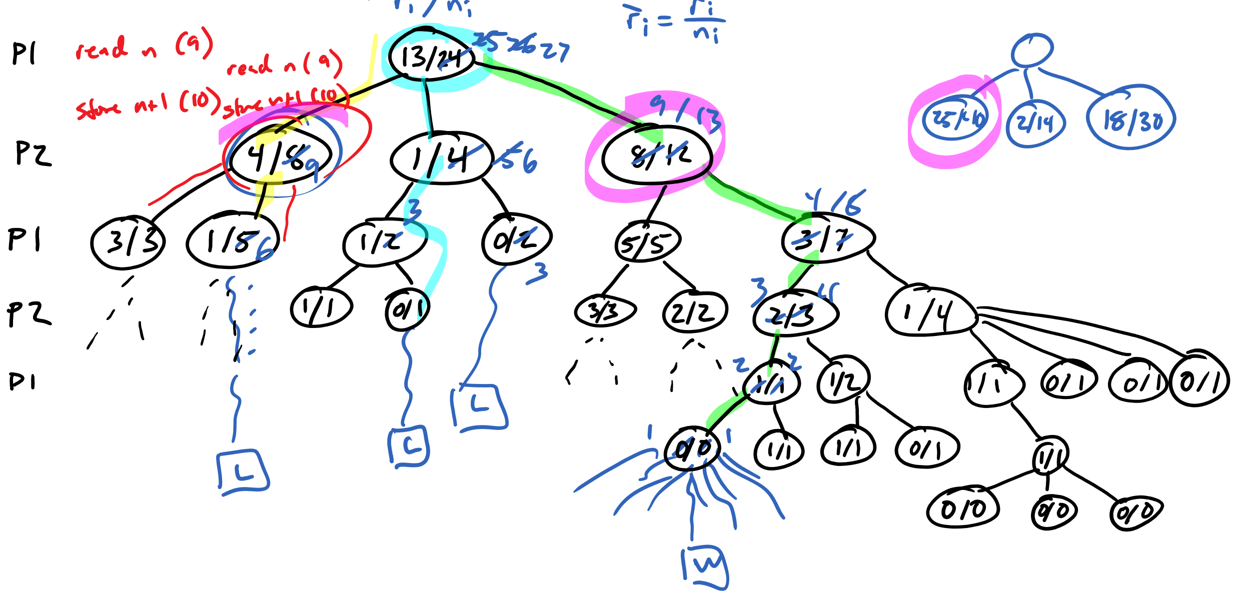
return move to child w/ best stats (highest mean reward or highest visit count)

UCT = MCTS + UCB




total reward → $r_i / n_i$ ← number of visits

$\bar{r}_i = \frac{r_i}{n_i}$




UCT = MCTS

advantages: convergent — converges to minimax (given enough time)

anytime — returns a move at any point

no domain knowledge — no heuristic needed

easily parallelized — leaf – multiple parallel playouts from leaves

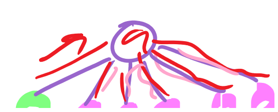
tree – parallel traversals through tree

root – separate tree for each thread; combine results (majority rule combining stats)

disadvantages: no domain knowledge

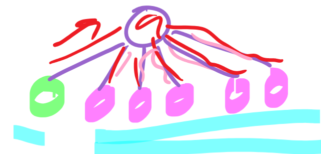
some games not amenable

trap state: most moves bad

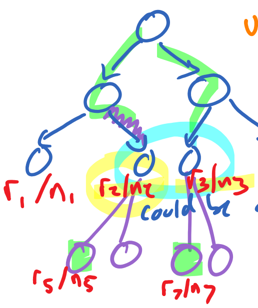


some games not amenable

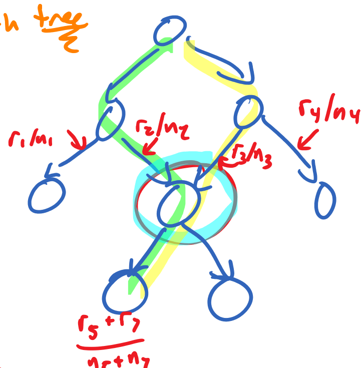
trap state: most moves bad
one good move



MCTS adapted for games that aren't trees

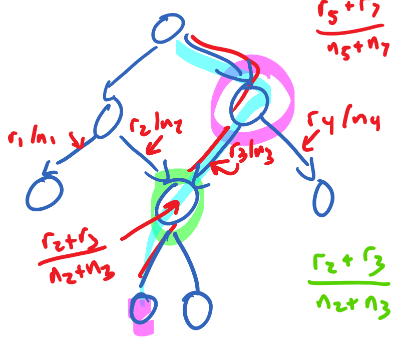



UCB0: normal MCTS with tree




UCB1: merge nodes but use stats on outgoing edges

implicit tree: edges not stored anywhere

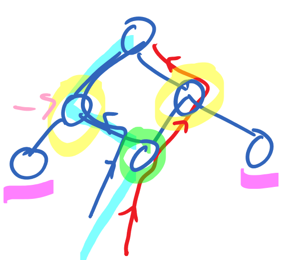



UCB3: backprop along all paths leaf→root instead of one you came down

UCB2: combines observed reward over all paths to node (exploit)

not for exploration term

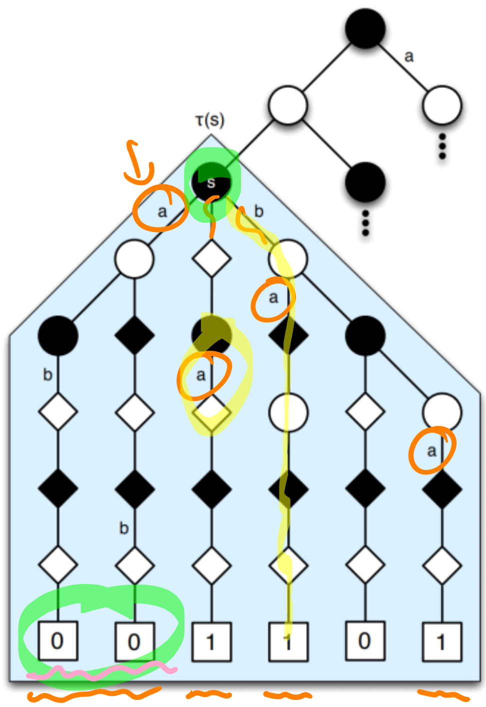
$$\frac{r_2+r_3}{n_2+n_3} + \sqrt{\frac{2\ln\boxed{n_3+n_4}}{n_3}}$$



$$\frac{r_2+r_3}{n_2+n_3} + \cancel{\sqrt{\frac{2\ln\boxed{(n_2+n_3)+n_4}}{n_2+n_3}}}$$

breaks convergence

**MC-RAVE** rapid averaging value estimation




$$Q(s,a) = 0/2$$
$$Q(s,b) = 2/3$$
$$\tilde{Q}(s,a) = 3/5$$
$$\tilde{Q}(s,b) = 2/5$$

all moves as first (AMAF)

$Q(s,a)$ : observed reward of action a in state s $= \frac{0}{2}$

$\tilde{Q}(s,a)$ : uses AMAF heuristic obs reward for a over entire subtree rooted at s

From Gelly and Silver, Monte-Carlo tree search and rapid action value estimation in computer Go. Artif. Intell. 175, 1856-1875, 2011

weight of $Q(s,a)$ vs $\tilde{Q}(s,a)$

$$Q_*(s,a) = (1 - \underbrace{\beta(s,a)}_{\sqrt{\frac{k}{3N(s)+k}}})\, Q(s,a) + \beta(s,a)\tilde{Q}(s,a)$$

$\sqrt{\frac{k}{3N(s)+k}}$ for Go ($k \approx 1000$) Fig 5

$N(s)$: # visits for s