Temperature Learning in MCTS

Kuo-Yuan Kao
National Penghu University

Reinforcement Learning

Model: (MDP)

- S: a set of states
- A: a set of actions
- T: stochastic transition from $S \times A$ to S
- r: immediate reward of an action from a state
- π: policy, a stochastic mapping from S to A.

Agent goal:

find a good policy to maximize long term reward

Monte Carlo Tree Search

Idea:

- evaluate the action-value online from simulations
- use values in the search tree to improve policy

$$Q^{\pi}(s,a) = E_{\pi}[z \mid s_t = s, a_t = a]$$

Upper Confident Tree

Idea:

- Balancing exploitation & exploration

$$Q^{\oplus}(s,a) = Q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}}$$

$$a^* = \arg_a \max Q^{\oplus}(s,a)$$

Rapid Action Value Evaluation

Idea:

- one general value for each move, regardless of when it is played.

$$\tilde{Q}^{\pi}(s,a) = E_{\pi}[z \mid s_t = s, \exists u > t \quad s.t.\ a_u = a]$$

Tuning (tuning and tuning …)

Idea:

- Experiment result showing it works

$$\hat{Q}(s,a) = (1-\beta) \times Q(s,a) + \beta \times \tilde{Q}(s,a)$$

Task of MCTS

- How to learn default policy online?

MCTS for Combinatorial Games

Model: (MCTS-CG)

- The global state is a Cartesian product of local states.
- The global action set is a union of local action sets; each local action can only change the local state.
- The global reward is a sum of local rewards; each local reward depends on the corresponding local state.

$$S = \prod_k S_k$$

$$A = \bigcup_k A_k$$

$$R(s \in S) = \sum_k R_k(s_k)$$

Mean and Temperature

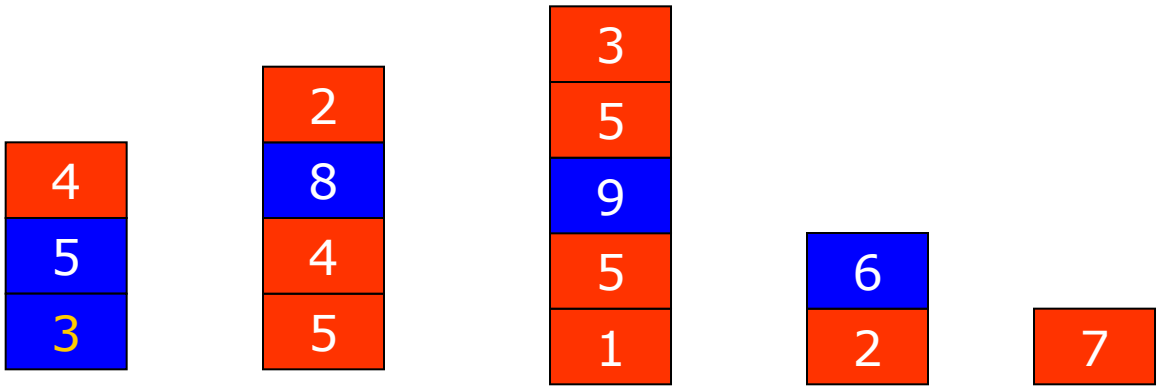
- Each game has a mean and temperature
- These values of a game are *independent* of the states of other games.
- Simply playing the game with maximum temperature can result *close to optimal* outcome.

- In the other way,

An optimal move usually has …

the maximum temperature.

Example: Heap Go



- The complexity to calculate the mt-value of a Heap-Go heap is O(n2),
  where n is the number of counters in the heap.

The Task

- How can the agent learn the temperature of a local state in MCTS-CG?

The Idea

- Each simulation carries a lot of information, the agent should utilize the information as much as possible.
- Given Q(s, a)=10, Q(s, b)=2, the agent learns
  (1) the expected outcome of a is better than the one of b.
  (2) there is *great chance* that the *temperature* of a is higher than the one of b.
- There is no need to know the exact temperatures of local states, all the agent need to know which local state has the *relative* maximum temperature.

Action Reward Difference

- Within the backup operator of a simulation, learn the difference between action rewards.

$$D(a,b) = \mathrm{E}[Q(s,a) - Q(s,b) \mid s \in S], where \quad a,b \in A$$

- The action difference implies a relation

$$a \succ b \Leftrightarrow D(a,b) > 0$$

- In terms of CG, a has a higher temperature than b's

Action Priority

- Let *AP*= {a_1, …, a_n} be a permutation of actions. Define

$$AP^* = \arg_{AP} \min Cost(AP, D)$$

where

$$Cost(AP, D) = \sum_{i>j} D^+(a_i, a_j)$$

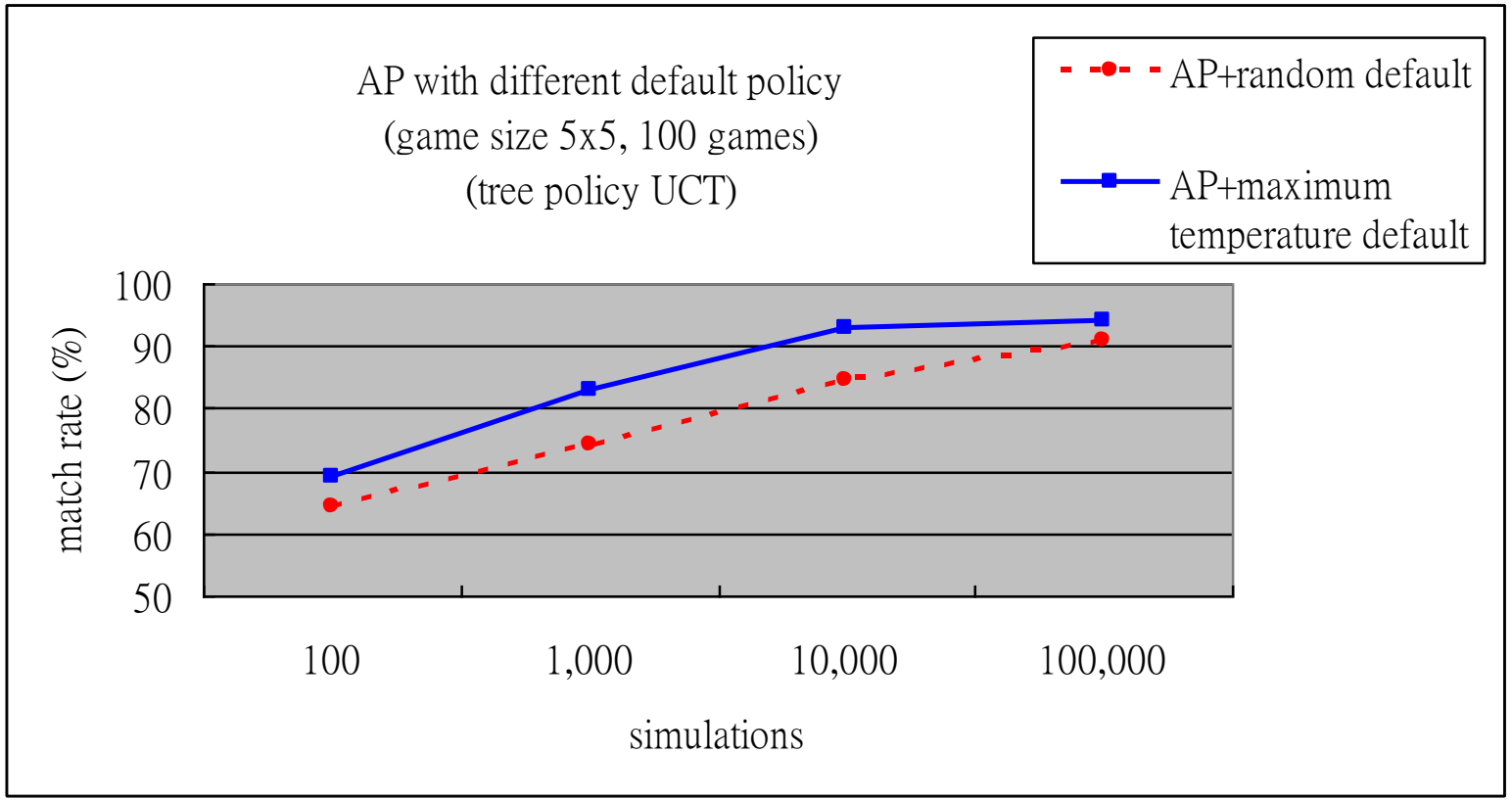
*AP** is the optimal action permutation.

- *AP** can be learned incrementally. (detailed in my paper)

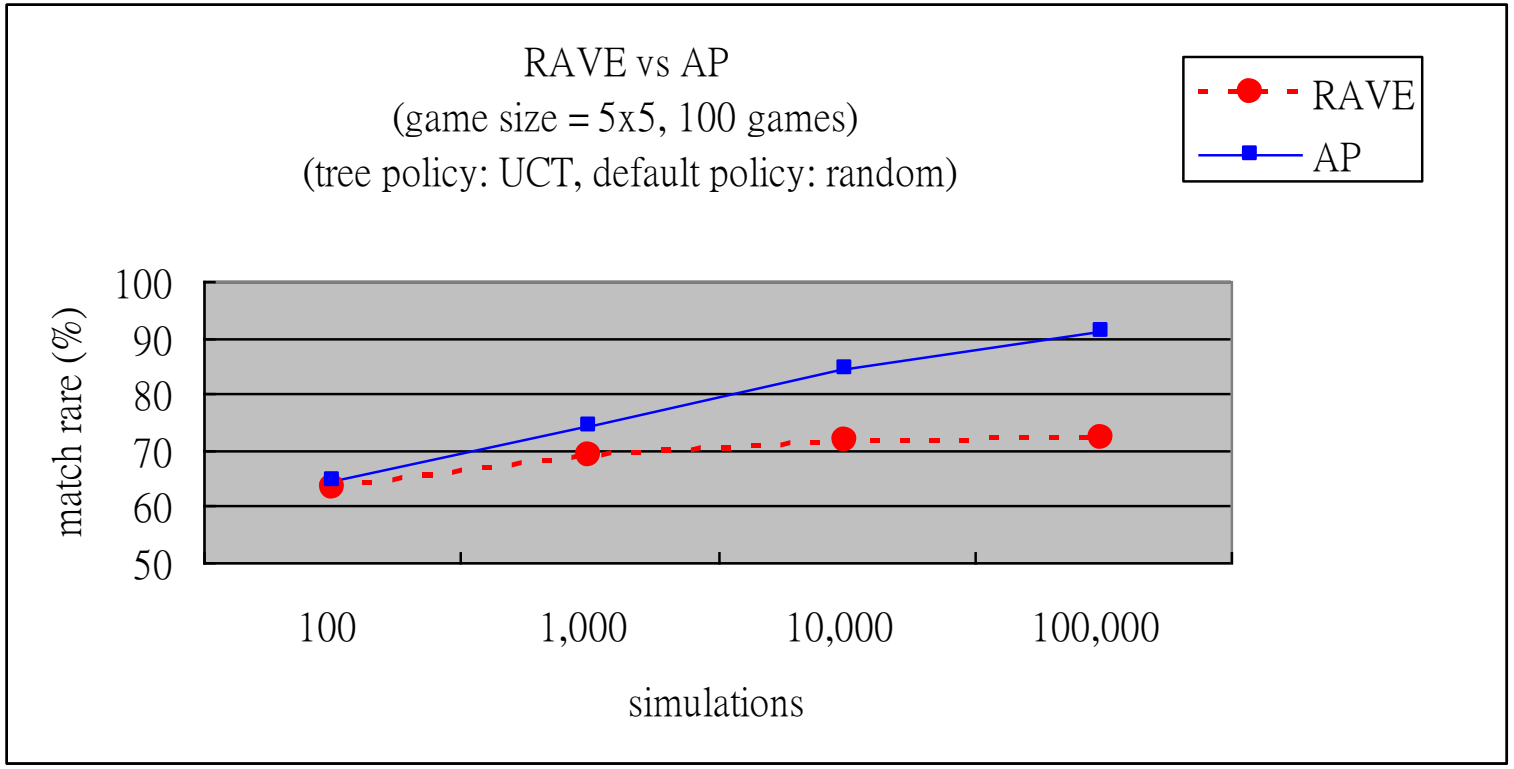
Match Rate

- A number between 0 and 1 which measures of the order difference between two permutations of a set.
- Match rate 1 means the orders are completely match.

Experiment Result (1)



Experiment Result (2)



Conclusion

- Very positive result of temperature learning
- AP can be used to guide the default policy
- Using temperature list as the target, AP has a higher match rate than RAVE.

temperature= gradient

Further Consideration

- Local temperature: local APs can be combined into one global AP. (This will significantly reduce the search space.)
- True temperature learning in MCTS.
- Learning the mean of a game.
- Application to computer Go and other games.
- Generalization to general RL problems.
- Generalization to mathematical optimization with objective function of the form:

$$f(x_1,\ldots,x_n)=\sum_k f_k(x_k)$$

- Application to energy stock problem, stochastic linear programming...

Other idea?

We need killer application other than computer games.