Monte-Carlo Tree Search
An introduction

Jérémie DECOCK

Inria Saclay - LRI

May 2012
Introduction
**Monte-Carlo Tree Search (MCTS)**

- MCTS is a recent algorithm for *sequential decision making*.
- It applies to *Markov Decision Processes* (MDP):
  - discrete-time $t$ with finite horizon $T$
  - state $s_t \in S$
  - action $a_t \in A$
  - transition function $s_{t+1} = \mathcal{P}(s_t, a_t)$
  - cost function $r_t = R\mathcal{P}(s_t)$
  - reward $R = \sum_{t=0}^{T} r_t$
  - policy function $a_t = \pi\mathcal{P}(s_t)$
- we look for the policy $\pi^*$ that maximizes expected $R$
MCTS strength

- Mcts is a versatile algorithm (it does not require knowledge about the problem)
- especially, does not require any knowledge about the Bellman value function
- stable on high dimensional problems
- it outperforms all other algorithms on some problems (difficult games like Go, general game playing, . . .)
MCTS

Problem are represented as a tree structure:

- blue circles = states
- plain edges + red squares = decisions
- dashed edges = stochastic transitions between two states
Monte-Carlo Tree Search
Main steps of MCTS

Monte-Carlo Tree Search

Selection and expansion

References

Decock Inria Saclay - LRI

Monte-Carlo Tree Search
Main steps of MCTS

Starting from an initial state:

1. select the state we want to expand from
Main steps of MCTS

Starting from an initial state:

1. select the state we want to expand from
2. add the generated state in memory
Main steps of MCTS

Starting from an initial state:

1. select the state we want to expand from
2. add the generated state in memory
3. evaluate the new state with a default policy until horizon is reached
Main steps of MCTS

Starting from an initial state:

1. select the state we want to expand from
2. add the generated state in memory
3. evaluate the new state with a default policy until horizon is reached
4. back-propagation of some information:
   a. $n(s, a)$: number of times decision $a$ has been simulated in $s$
   b. $n(s)$: number of time $s$ has been visited in simulations
   c. $\hat{Q}(s, a)$: mean reward of simulations where $a$ was chosen in $s$
Main steps of MCTS

Starting from an initial state:

1. select the state we want to expand from
2. add the generated state in memory
3. evaluate the new state with a default policy until horizon is reached
4. back-propagation of some information:
   - $n(s, a)$ : number of times decision $a$ has been simulated in $s$
   - $n(s)$ : number of times state $s$ has been visited in simulations
   - $\hat{Q}(s, a)$ : mean reward of simulations where $a$ was chosen in $s$
Main steps of MCTS

1. selection
2. expansion
3. simulation
4. propagation

The selected decision

\[ a_{t_n} = \text{the most visited decision form the current state (root node)} \]
Selection and expansion
Selection step

How to select the state to expand?
How to select the state to expand?

The selection phase is driven by Upper Confidence Bound

$$\text{score}_{ucb}(s, a) = \hat{Q}(s, a) + \sqrt{\frac{\log(2 + n(s))}{2 + n(s, a)}}$$

1. mean reward of simulations including action $a$ in state $s$
2. the uncertainty on this estimation of the action’s value
How to select the state to expand?

The selection phase is driven by **Upper Confidence Bound**

$$\text{score}_{ucb}(s, a) = \hat{Q}(s, a) + \sqrt{\frac{\log(2 + n(s))}{2 + n(s, a)}}$$

The selected action:

$$a^* = \arg\max_a \text{score}_{ucb}(s, a)$$
How to select the state to expand?

\[ \sqrt{\frac{\log(2+n(s))}{2+n(s,a)}} \]
When should we expand?

One standard way of tackling the exploration/exploitation dilemma is *Progressive Widening*.

A new parameter $\alpha \in [0; 1]$ is introduced, to choose between exploration (add a decision to the tree) and exploitation (go to an existing node).
How to select the state to expand?

- if($|\mathcal{A}'_s| < n(s)^{\alpha}$) then we explore a new decision
- else we simulate a known decision

With $|\mathcal{A}'_s|$ the number of legal actions in state $s$
When should we expand?

$\alpha = 0.2$

$\alpha = 0.8$
References I