Selection and expansion 000000

Image: A math and A

## Monte-Carlo Tree Search An introduction

#### Jérémie DECOCK

Inria Saclay - LRI

May 2012

Decock

Monte-Carlo Tree Search

Inria Saclay - LRI

Selection and expansion 000000

(日)

2

Inria Saclay - LRI

# Introduction

Decock

3

Inria Saclav - LRI

## 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  $\mathbf{s}_t \in \mathcal{S}$
  - ▶ action  $\mathbf{a}_t \in \mathcal{A}$
  - transition function  $\mathbf{s}_{t+1} = \mathcal{P}(\mathbf{s}_t, \mathbf{a}_t)$
  - cost function  $r_t = \mathcal{R}_{\mathcal{P}}(\mathbf{s}_t)$
  - reward  $R = \sum_{t=0}^{T} r_t$
  - policy function  $\mathbf{a}_t = \pi_{\mathcal{P}}(\mathbf{s}_t)$
  - we look for the policy  $\pi^*$  that maximizes expected R

## MCTS strength

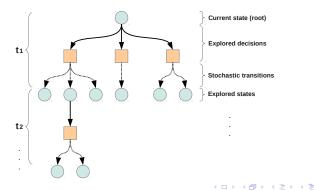
- Mcts is a versatile algorithm (it does not require knowledge about the problem)
- especially, does not require any knowledge about the Bellman value function
- stable on high dimensional problems
- it outperforms all other algorithms on some problems (difficult games like Go, general game playing,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

Introduction 00●	<i>Monte-Carlo Tree Search</i> 000	Selection and expansion	References
Introduction			

# MCTS

Problems are represented as a tree structure:

- blue circles = states
- plain edges + red squares = decisions
- dashed edges = stochastic transitions between two states



Inria Saclav - LRI

Introduction 000 Monte-Carlo Tree Search

Selection and expansion

Image: Image:

-

References

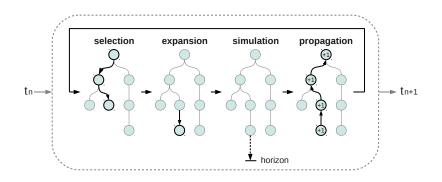
6

Inria Saclay - LRI

# Monte-Carlo Tree Search

Decock

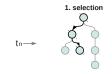
Introduction 000	Monte-Carlo Tree Search ●00	Selection and expansion	References
Monte-Carlo Tree Search			



7

Decock

Introduction 000	Monte-Carlo Tree Search ○●○	Selection and expansion	References
Monte-Carlo Tree Search			
Main steps of	MCTS		



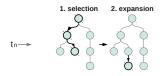
--> tn+1

Starting from an initial state:

 $1. \ \mbox{select the state we want to expand from}$ 

Decock

Introduction 000	Monte-Carlo Tree Search ○●○	Selection and expansion	References
Monte-Carlo Tree Search			



—> tn+1

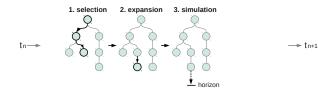
8

Inria Saclay - LRI

Starting from an initial state:

- 1. select the state we want to expand from
- 2. add the generated state in memory

Introduction 000	Monte-Carlo Tree Search 0●0	Selection and expansion	References
Monte-Carlo Tree Search			



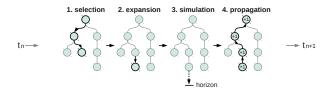
Starting from an initial state:

- $1. \ \mbox{select the state we want to expand from}$
- 2. add the generated state in memory
- 3. evaluate the new state with a default policy until horizon is reached

8

Inria Saclav - LRI

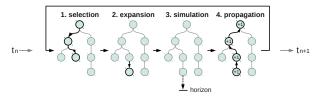
Introduction 000	Monte-Carlo Tree Search ○●○	Selection and expansion	References
Monte-Carlo Tree Search			



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:
  - $n(\mathbf{s}, \mathbf{a})$  : number of times decision  $\mathbf{a}$  has been simulated in  $\mathbf{s}$
  - $n(\mathbf{s})$ : number of time **s** has been visited in simulations
  - $\hat{Q}(\mathbf{s}, \mathbf{a})$ : mean reward of simulations where  $\mathbf{a}$  was whosen in  $\mathbf{s}$

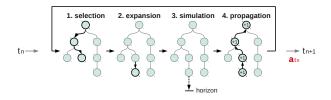
Introduction 000	Monte-Carlo Tree Search ○●○	Selection and expansion	References
Monte-Carlo Tree Search			



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:
  - $n(\mathbf{s}, \mathbf{a})$  : number of times decision  $\mathbf{a}$  has been simulated in  $\mathbf{s}$
  - $n(\mathbf{s})$  : number of time **s** has been visited in simulations
  - $\hat{Q}(\mathbf{s}, \mathbf{a})$ : mean reward of simulations where  $\mathbf{a}$  was whosen in  $\mathbf{s}$

Introduction 000	Monte-Carlo Tree Search ○○●	Selection and expansion	References
Monte-Carlo Tree Search			



#### The selected decision

 $\mathbf{a}_{t_n}$  = the most visited decision form the current state (root node)

Inria Saclay - LRI

Introduction 000 Monte-Carlo Tree Search

Selection and expansion

References

10

Inria Saclay - LRI

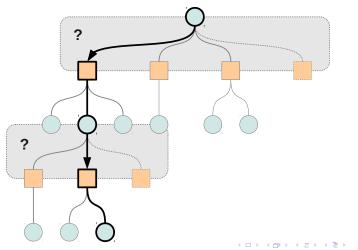
# Selection and expansion

Decock

Introduction 000	Monte-Carlo Tree Search	Selection and expansion •00000	References
Selection and expansion			

#### Selection step

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



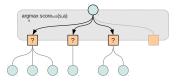
11

Inria Saclay - LRI

Decock

Introduction 000	<i>Monte-Carlo Tree Search</i> 000	Selection and expansion	References
Selection and expansion			

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



The selection phase is driven by Upper Confidence Bound

$$\text{score}_{\text{ucb}}(\mathbf{s}, \mathbf{a}) = \underbrace{\hat{Q}(\mathbf{s}, \mathbf{a})}_{1} + \underbrace{\sqrt{\frac{\log(2 + n(\mathbf{s}))}{2 + n(\mathbf{s}, \mathbf{a})}}}_{2}$$

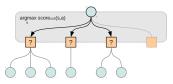
- 1. mean reward of simulations including action  $\mathbf{a}$  in state  $\mathbf{s}$
- 2. the uncertainty on this estimation of the action's value

Decock

Inria Saclav - LRI

Introduction 000	<i>Monte-Carlo Tree Search</i> 000	Selection and expansion	References
Selection and expansion			

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



The selection phase is driven by Upper Confidence Bound

$$score_{ucb}(\mathbf{s}, \mathbf{a}) = \underbrace{\hat{Q}(\mathbf{s}, \mathbf{a})}_{1} + \underbrace{\sqrt{\frac{\log(2 + n(\mathbf{s}))}{2 + n(\mathbf{s}, \mathbf{a})}}}_{2}$$

The selected action:

$$\mathbf{a}^{\star} = \arg \max_{\mathbf{a}} \operatorname{score}_{\operatorname{ucb}}(\mathbf{s}, \mathbf{a})$$

12

Inria Saclav - LRI

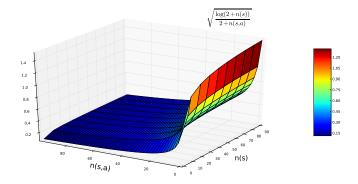
Decock

Monte-Carlo Tree Search

Selection and expansion 000000

Selection and expansion

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

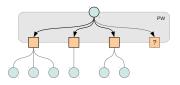




Decock

Introduction 000	<i>Monte-Carlo Tree Search</i> 000	Selection and expansion	References
Selection and expansion			

#### When should we expand?



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

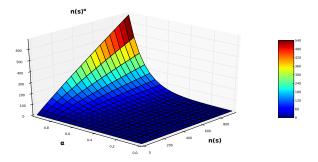
A new parameter  $\alpha \in [0; 1]$  is introduced, to choose between exploration (add a decision to the tree) and exploitation (go to an existing node)

14

Inria Saclav - LRI

Decock

Introduction 000	Monte-Carlo Tree Search 000	Selection and expansion



15

Inria Saclav - LRI

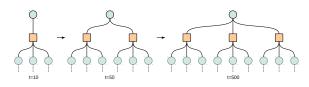
- if  $(|\mathcal{A}'_{\mathbf{s}}| < n(\mathbf{s})^{\alpha})$  then we explore a new decision
- else we simulate a known decision
- With  $|\mathcal{A}'_{\mathbf{s}}|$  the number of legal actions in state  $\mathbf{s}$

Decock

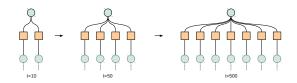
Introduction	<i>Monte-Carlo Tree Search</i>	Selection and expansion	References
000	000	00000●	
Selection and expansion			

### When should we expand?

 $\alpha = 0.2$ 



 $\alpha = \mathbf{0.8}$ 



Inria Saclay - LRI

16

Decock

Introduction 000 Monte-Carlo Tree Search

Selection and expansion

References

#### References I



Decock