Planning and Optimization

G5. Monte-Carlo Tree Search: Framework

Gabriele Röger and Thomas Keller

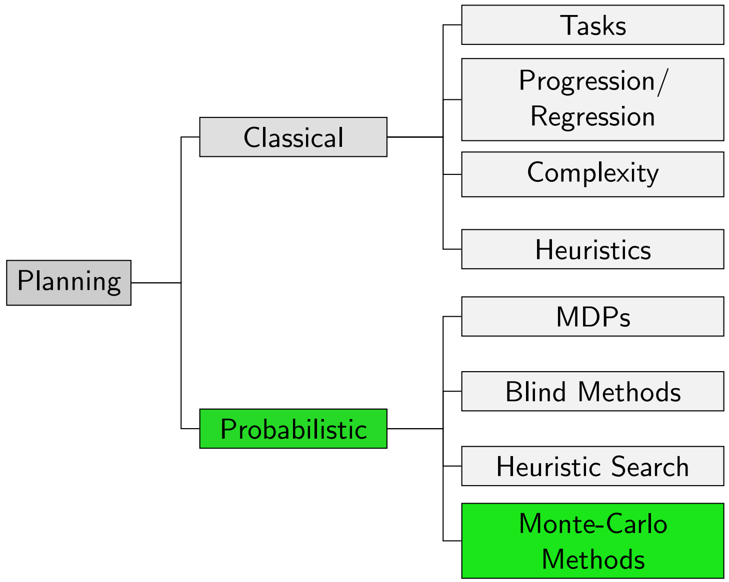
Universität Basel

December 10, 2018

Planning and Optimization

December 10, 2018 — G5. Monte-Carlo Tree Search: Framework

Content of this Course



G5.1 Motivation

Motivation

- Discussed Monte-Carlo methods asymptotically suboptimal
- Some members of Monte-Carlo Tree Search (MCTS) framework asymptotically optimal
- Have already seen what Monte-Carlo means
  $\Rightarrow$ we only consider algorithms that perform Monte-Carlo samples and use Monte-Carlo backups as MCTS
- Difference to previous methods: tree search

G5.2 MCTS Tree

MCTS Tree

- Like RTDP, MCTS performs trials (or rollouts)
- Like AO*, MCTS iteratively builds explicit representation of SSP
- MCTS explicates SSP (or MDP) as search tree
- Duplicates (also: transposition) possible, i.e., multiple search nodes with identical associated state
- Search tree can have unbounded depth

Tree Structure

- Differentiate between two types of search nodes:
  - Decision or OR nodes
  - Chance or AND nodes
- Search nodes correspond 1:1 to traces from initial state
- Decision and chance nodes alternate
- Decision nodes correspond to states in a trace
- Chance nodes correspond to actions (labels) in a trace
- Decision nodes have (up to) one child node for each applicable action
- Chance nodes have (up to) one child node for each outcome

AND/OR Tree

Definition (AND/OR Tree)

An AND/OR tree is given by a tuple $\mathcal{G} = \langle d_0, D, C, E \rangle$, where

- D and C are disjunct sets of decision and chance nodes
- $d_0 \in D$ is the root node
- $E \subseteq (D \times C) \cup (C \times D)$ is the set of edges such that the graph $\langle D \cup C, E \rangle$ is a tree

Search Node Annotations

- Decision nodes d are annotated with
  - visit counter $N(d)$
  - state-value estimate $\hat{V}(d)$
  - state $s(d)$
  - probability $p(d)$
- Chance nodes c are annotated with
  - visit counter $N(c)$
  - action-value (or Q-value) estimate $\hat{Q}(c)$
  - state $s(c)$
  - action $a(c)$
- With children(n), we refer to explicated child nodes of node n

Note: states, actions and probabilities can often be computed on the fly

AND/OR Tree over SSP

Definition (AND/OR Tree)

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$ be an SSP. An AND/OR tree $\mathcal{G} = \langle d_0, D, C, E \rangle$ is an AND/OR tree over $\mathcal{T}$ if

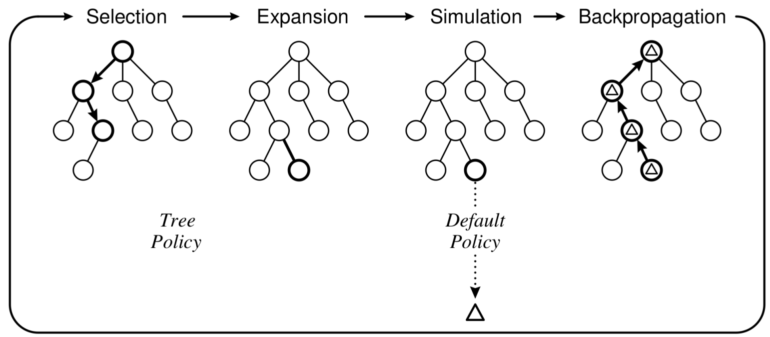
- $s(d_0) = s_0$
- $s(n) \in S$ for all $n \in C \cup D$
- $\langle d, c \rangle \in E$ for $d \in D$ and $c \in C$ iff $s(c) = s(d)$ and $a(c) \in L(s(c))$
- $\langle d, c \rangle \in E$ and $\langle d, c' \rangle \in E \Rightarrow c = c'$ or $a(c) \neq a(c')$
- $\langle c, d \rangle \in E$ for $c \in C$ and $d \in D$ iff $T(s(c), a(c), s(d)) > 0$ and $p(d) = T(s(c), a(c), s(d))$
- $\langle c, d \rangle \in E$ and $\langle c, d' \rangle \in E \Rightarrow d = d'$ or $s(d) \neq s(d')$

G5.3 Framework

Trials

- The search tree is build in trials
- Trials are performed as long as resources (deliberation time, memory) allow
- Initially, the search tree consist of only the root node
- Trials (may) add search nodes to the tree
- Search tree at the end of the i-th trial denoted with $\mathcal{G}^i$
- Use same superscript for annotations of search nodes (visit counter and state- and action-value estimates)

Trials



Taken from Browne et al., "A Survey of Monte Carlo Tree Search Methods", 2012

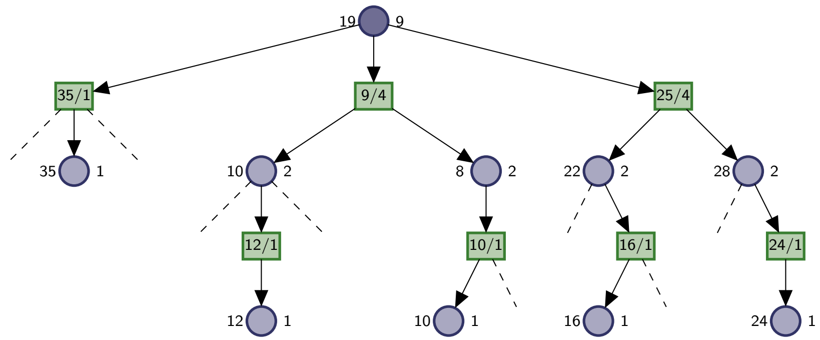
Phases of Trials

Each trial consists of (up to) four phases:

- Selection: traverse the tree by sampling the execution of the tree policy until
  1. an action is applicable that is not explicated, or
  2. an outcome is sampled that is not explicated, or
  3. a goal state is reached
- Expansion: create search nodes for the applicable action and a sampled outcome (case 1) or just the outcome (case 2)
- Simulation: sample default policy until a goal state is reached
- Backpropagation: update each visited node by
  - extending average state-/action-values estimate with accumulated cost following the search node (both from simulation and decisions in the tree)
  - increasing visit counter by 1

MCTS: Example

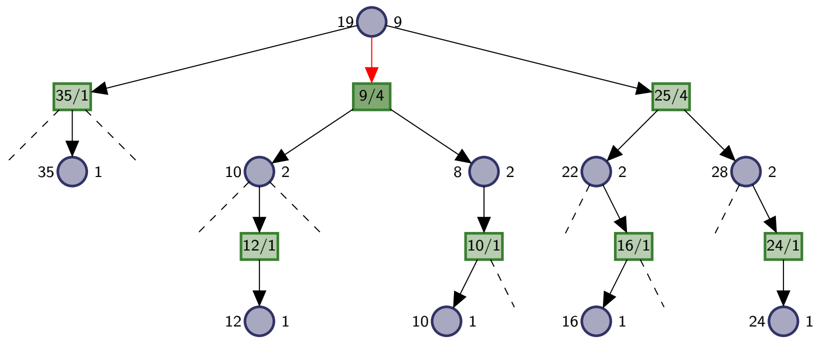
Selection phase: apply tree policy to traverse tree



(for simplicity, all costs in the tree are 0)

MCTS: Example

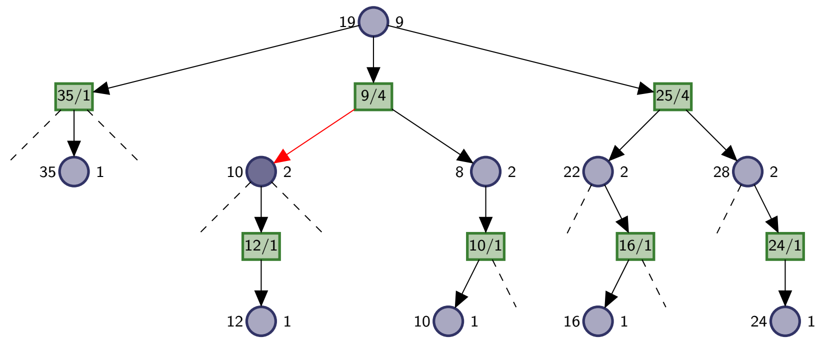
Selection phase: apply tree policy to traverse tree



(for simplicity, all costs in the tree are 0)

MCTS: Example

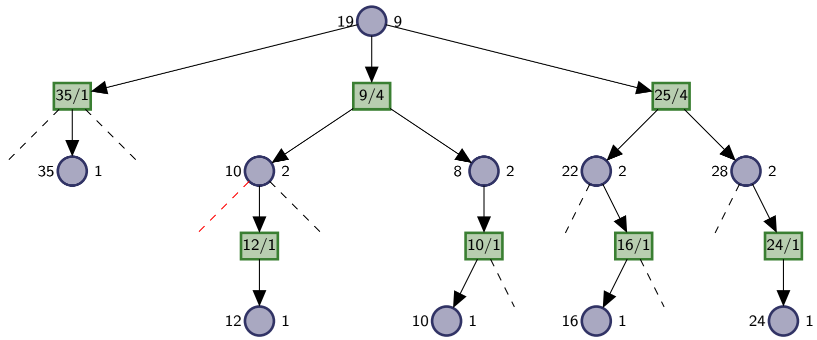
Selection phase: apply tree policy to traverse tree



(for simplicity, all costs in the tree are 0)

MCTS: Example

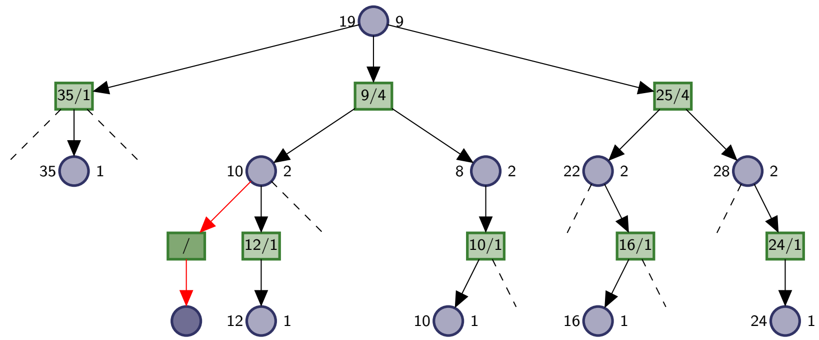
Selection phase: apply tree policy to traverse tree



(for simplicity, all costs in the tree are 0)

MCTS: Example

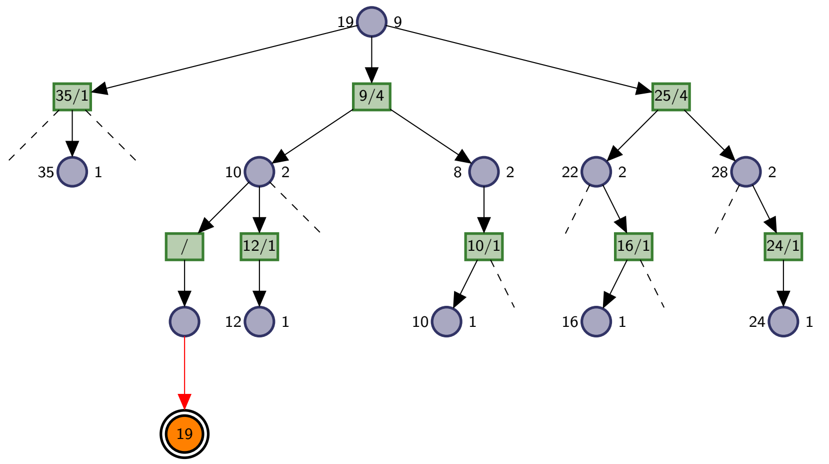
Expansion phase: create search nodes



(for simplicity, all costs in the tree are 0)

MCTS: Example

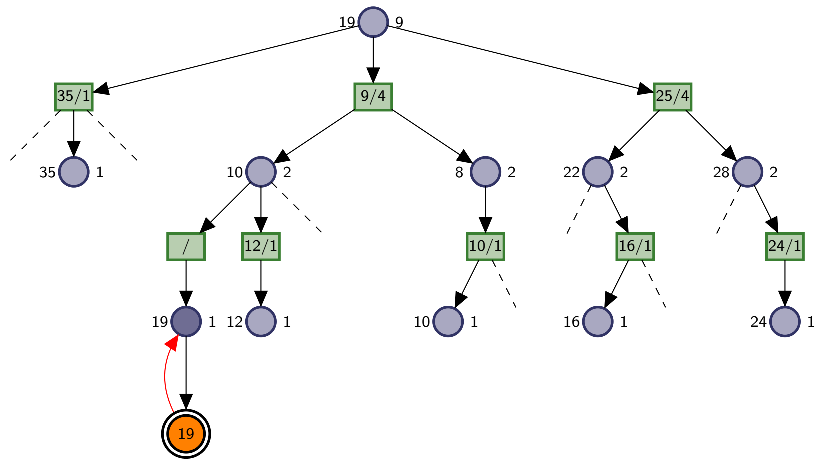
Simulation phase: apply default policy until goal



(for simplicity, all costs in the tree are 0)

MCTS: Example

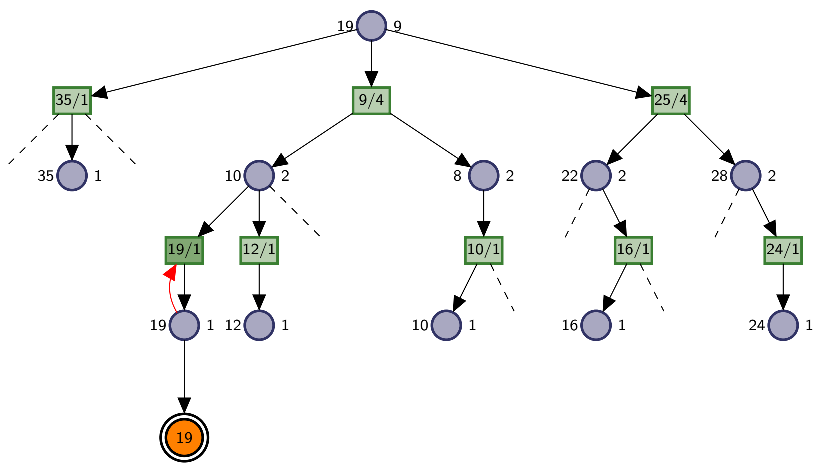
Backpropagation phase: update visited nodes



(for simplicity, all costs in the tree are 0)

MCTS: Example

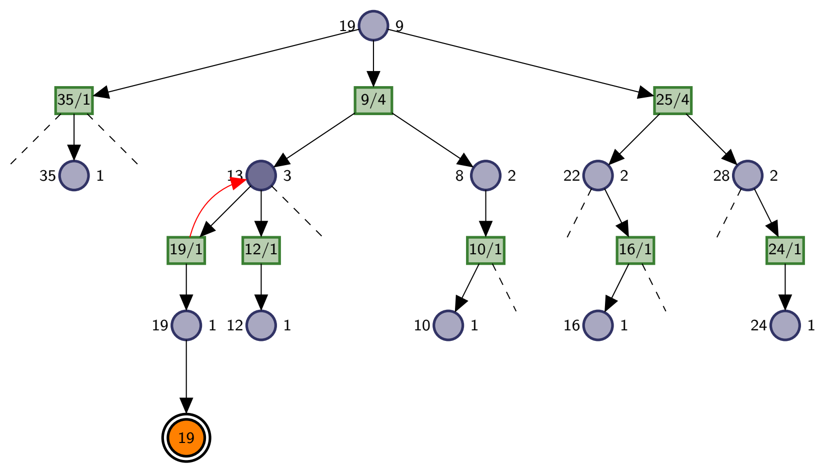
Backpropagation phase: update visited nodes



(for simplicity, all costs in the tree are 0)

MCTS: Example

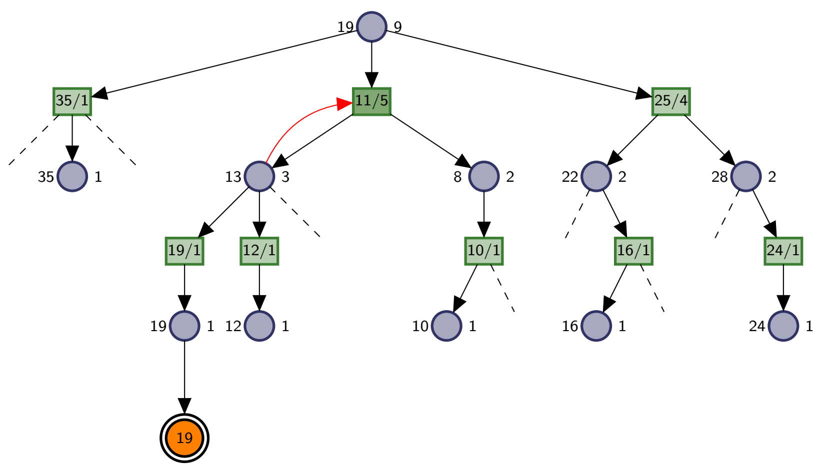
Backpropagation phase: update visited nodes



(for simplicity, all costs in the tree are 0)

MCTS: Example

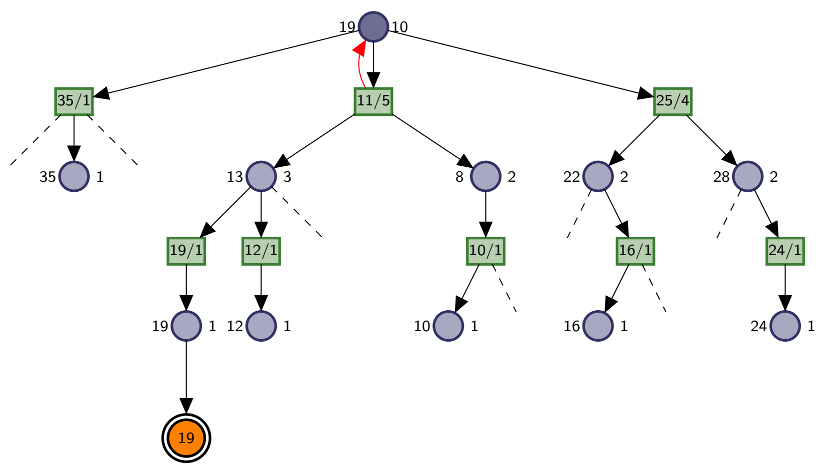
Backpropagation phase: update visited nodes



(for simplicity, all costs in the tree are 0)

MCTS: Example

Backpropagation phase: update visited nodes



(for simplicity, all costs in the tree are 0)

MCTS Framework

Member of MCTS framework are specified in terms of:

- Tree policy
- Default policy

MCTS Tree Policy

Definition (Tree Policy)
Let $\mathcal{T}$ be an SSP. An MCTS tree policy is a probability distribution $\pi(a \mid d)$ over applicable actions $a \in L(s(d))$ for each decision node d.

Note: The tree policy (usually) takes information annotated in the current tree into account.

MCTS Default Policy

Definition (Default Policy)

Let $\mathcal{T}$ be an SSP. An MCTS default policy is a probability distribution $\pi(a \mid s)$ over applicable actions $a \in L(s)$ for each state $s \in S$.

Note: The default policy is independent of the search tree.

Monte-Carlo Tree Search

MCTS for SSP $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$

d_0 = create root node associated with s_0
while time allows:
  visit_decision_node($d_0, \mathcal{T}$)
return $a(\arg\min_{c \in \text{children}(d_0)} \hat{Q}(c))$

MCTS: Visit a Decision Node

visit_decision_node for decision node d, SSP $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$

if $s(d) \in S_\star$ **then return** 0
if there is $a \in L(s(d))$ not explicated:
  select such an a and add node c for $s(d), a$ to children(d)
else:
  c = tree_policy(d)
cost = visit_chance_node($c, \mathcal{T}$)
$\hat{V}(d) := \hat{V}(d) + \frac{\text{cost} - \hat{V}(d)}{N(d)+1}, N(d) := N(d) + 1$
return cost

MCTS: Visit a Chance Node

visit_chance_node for chance node c, SSP $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$

$s' \sim \text{succ}(s(c), a(c))$
let d be the node in children(c) with $s(d) = s'$
if there is no such node:
  add node d for s' to children(c)
  cost = sample_default_policy(s')
  $\hat{V}(d) := \text{cost}, N(d) := 1$
else:
  cost = visit_decision_node($d, \mathcal{T}$)
cost = cost + $c(s(c), a(c))$
$\hat{Q}(c) := \hat{Q}(c) + \frac{\text{cost} - \hat{Q}(c)}{N(c)+1}, N(c) := N(c) + 1$
return cost

G5.4 Summary

Summary

- Monte-Carlo Tree Search is a framework for algorithms
- MCTS algorithms perform trials
- Each trial consists of (up to) 4 phases
- MCTS algorithms are specified by a tree policy that describes behavior "in" tree
- and a default policy that describes behavior "outside" of tree