

# Planning and Optimization

## G3. Asymptotically Suboptimal Monte-Carlo Methods

Malte Helmert and Gabriele Röger

Universität Basel

December 9, 2020

# Planning and Optimization

December 9, 2020 — G3. Asymptotically Suboptimal Monte-Carlo Methods

## G3.1 Motivation

## G3.2 Monte-Carlo Methods

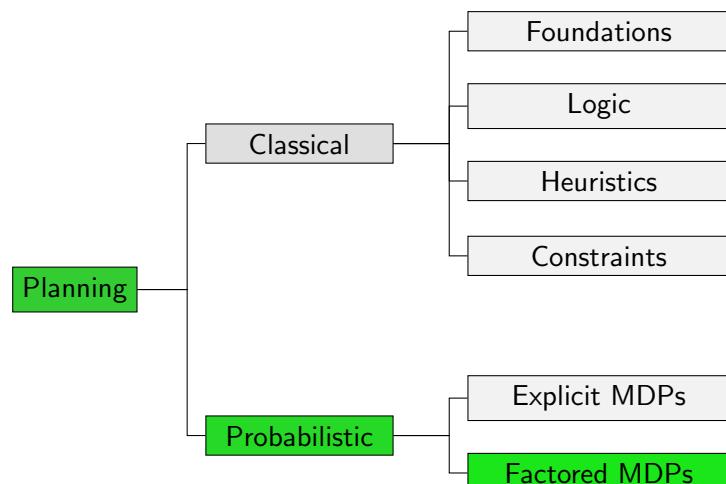
## G3.3 Hindsight Optimization

## G3.4 Policy Simulation

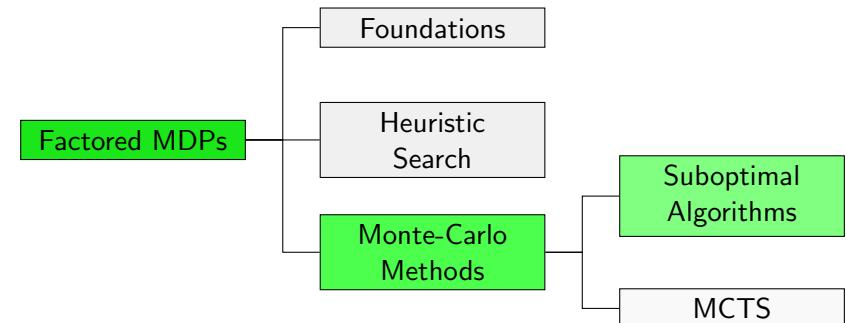
## G3.5 Sparse Sampling

## G3.6 Summary

## Content of this Course



## Content of this Course: Factored MDPs



## G3.1 Motivation

## Monte-Carlo Methods: Brief History

- ▶ 1930s: first researchers experiment with Monte-Carlo methods
- ▶ 1998: Ginsberg's GIB player competes with Bridge experts
- ▶ 2002: Kearns et al. propose Sparse Sampling
- ▶ 2002: Auer et al. present UCB1 action selection for multi-armed bandits
- ▶ 2006: Coulom coins term Monte-Carlo Tree Search (MCTS)
- ▶ 2006: Kocsis and Szepesvári combine UCB1 and MCTS to the famous MCTS variant, UCT
- ▶ 2007–2016: Constant progress of MCTS in Go culminates in AlphaGo's historical defeat of dan 9 player Lee Sedol

## G3.2 Monte-Carlo Methods

## Monte-Carlo Methods: Idea

- ▶ Summarize a broad family of algorithms
- ▶ Decisions are based on random samples (Monte-Carlo sampling)
- ▶ Results of samples are aggregated by computing the average (Monte-Carlo backups)
- ▶ Apart from that, algorithms can differ significantly

**Careful:** Many different definitions of MC methods in the literature

## Types of Random Samples

Random samples have in common that something is drawn from a given probability distribution. Some examples:

- ▶ a determinization is sampled (Hindsight Optimization)
- ▶ runs under a fixed policy are simulated (Policy Simulation)
- ▶ considered outcomes are sampled (Sparse Sampling)
- ▶ runs under an evolving policy are simulated (Monte-Carlo Tree Search)

## Reminder: Bellman Backups

Algorithms like Value Iteration or (L)RTDP use the Bellman equation as an update procedure.

The  $i$ -th state-value estimate of state  $s$ ,  $\hat{V}^i(s)$ , is computed with Bellman backups as

$$\hat{V}^i(s) := \min_{a \in A(s)} \left( c(a) + \sum_{s' \in S} T(s, a, s') \cdot \hat{V}^{i-1}(s') \right).$$

(Some algorithms use a heuristic if the state-value estimate on the right hand side of the Bellman backup is undefined.)

## Monte-Carlo Backups

Monte-Carlo methods instead estimate state-values by averaging over all samples.

Let  $N^i(s)$  be the number of samples for state  $s$  in the first  $i$  algorithm iterations and let  $cost^k(s)$  be the cost for  $s$  in the  $k$ -th sample ( $cost^k(s) = 0$  if the  $k$ -th sample has no estimate for  $s$ ).

The  $i$ -th state-value estimate of state  $s$ ,  $\hat{V}^i(s)$ , is computed with Monte-Carlo backups as

$$\hat{V}^i(s) := \frac{1}{N^i(s)} \cdot \sum_{k=1}^i cost^k(s).$$

## Monte-Carlo Backups: Properties

- ▶ no need to store  $cost^k(s)$  for  $k = 1, \dots, i$ : it is possible to compute Monte-Carlo backups iteratively as

$$\hat{V}^i(s) := \hat{V}^{i-1}(s) + \frac{1}{N^i(s)} (cost^i(s) - \hat{V}^{i-1}(s))$$

- ▶ no need to know SSP model for backups
- ▶ if  $s$  is a random variable,  $\hat{V}^i(s)$  converges to  $\mathbb{E}[s]$  due to the strong law of large numbers
- ▶ if  $s$  is not a random variable, this is not always the case

## G3.3 Hindsight Optimization

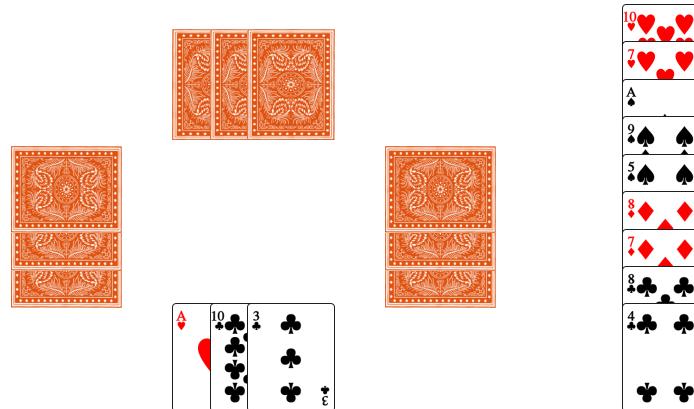
### Hindsight Optimization: Idea

Repeat as long as **resources** (deliberation time, memory) allow:

- ▶ **Sample** outcomes of all actions  
⇒ deterministic (classical) planning problem
- ▶ For each applicable action  $a \in A(s_0)$ ,  
compute **plan** in the sample that starts with  $a$

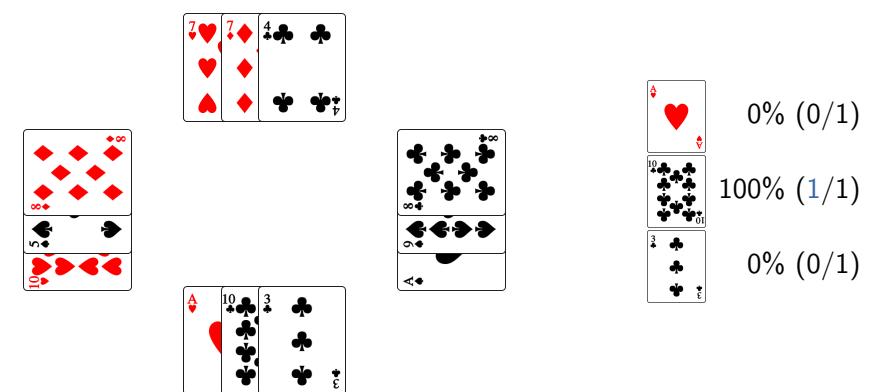
Execute the action with the **lowest average plan cost**

### Hindsight Optimization: Example



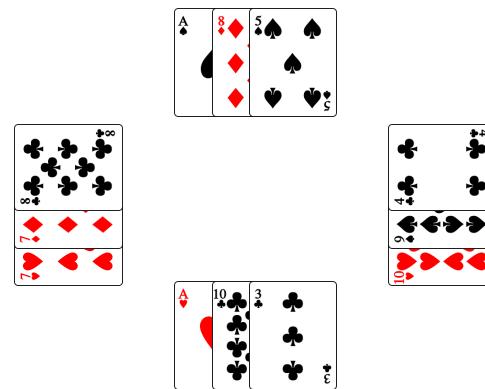
South to play, three tricks to win, trump suit ♣

### Hindsight Optimization: Example



South to play, three tricks to win, trump suit ♣

## Hindsight Optimization: Example

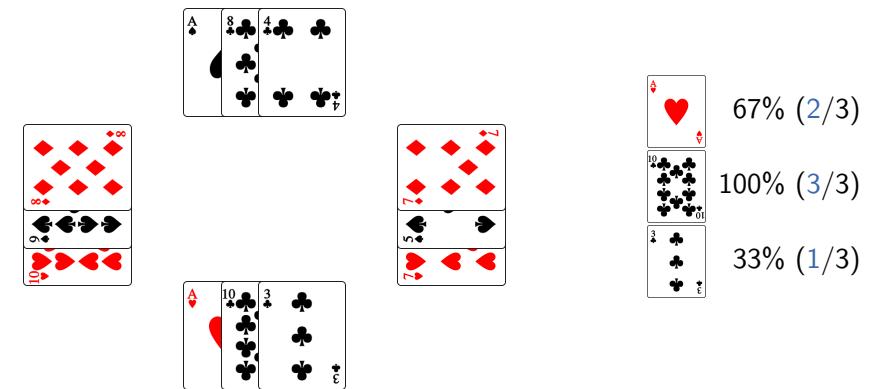


South to play, three tricks to win, trump suit ♣

## Hindsight Optimization: Evaluation

- ▶ HOP well-suited for some problems
- ▶ must be possible to solve sampled SSP efficiently:
  - ▶ domain-dependent knowledge (e.g., games like Bridge, Skat)
  - ▶ classical planner (FF-Hindsight, Yoon et. al, 2008)
- ▶ What about optimality in the limit?

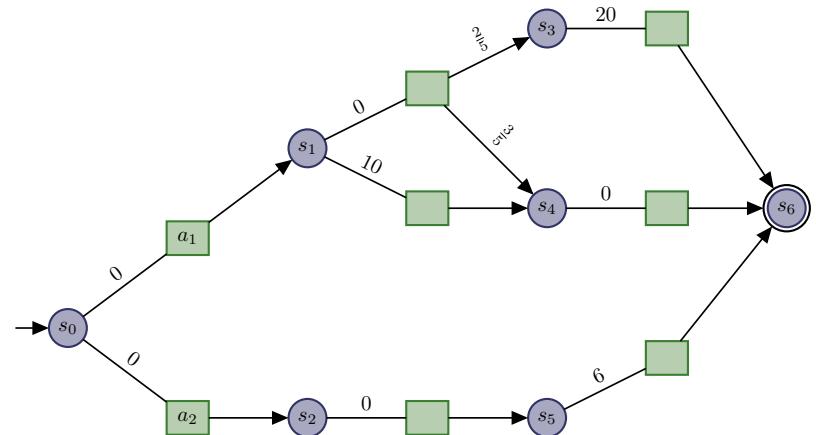
## Hindsight Optimization: Example



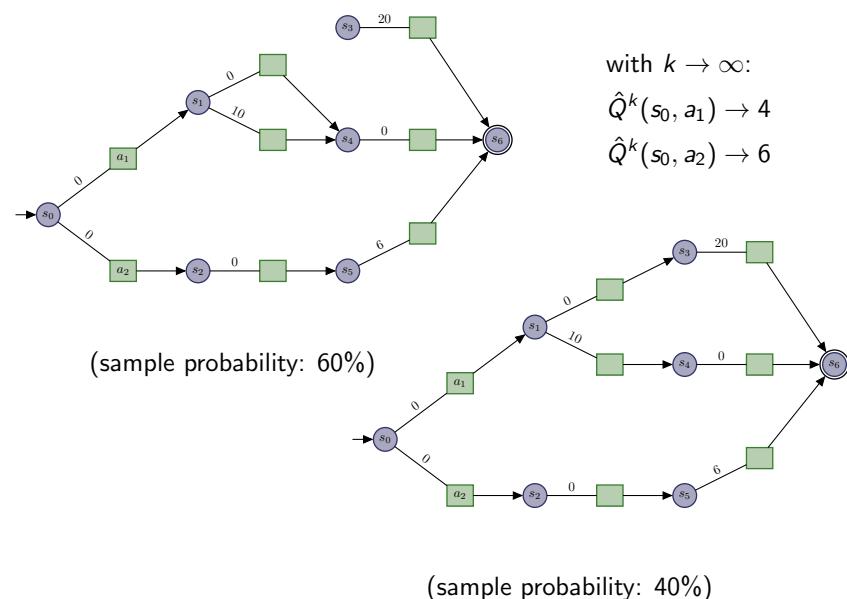
South to play, three tricks to win, trump suit ♣

## Hindsight Optimization: Evaluation

## Hindsight Optimization: Non-optimality in the Limit



## Hindsight Optimization: Non-optimality in the Limit



## Hindsight Optimization: Evaluation

- ▶ HOP **well-suited** for some problems
- ▶ must be possible to **solve** sampled MDP **efficiently**:
  - ▶ domain-dependent knowledge (e.g., games like Bridge, Skat)
  - ▶ classical planner (FF-Hindsight, Yoon et. al, 2008)
- ▶ What about optimality **in the limit**?  
 ⇒ in general not optimal due to **assumption of clairvoyance**

## G3.4 Policy Simulation

## Policy Simulation: Idea

Repeat as long as **resources** (deliberation time, memory) allow:

- ▶ For each applicable action  $a \in A(s_0)$ ,  
 start a **run** from  $s_0$  with  $a$  and then **follow a given policy  $\pi$**
- ▶ Execute the action with the **lowest average simulation cost**

Avoids clairvoyance by **evaluation** of policy through **simulation** of its execution.

## Policy Simulation: Evaluation

- ▶ Base policy is **static**
- ▶ No mechanism to **overcome** the weaknesses of base policy (if there are no weaknesses, we don't need policy simulation)
- ▶ **Suboptimal decisions** in simulation affect policy quality
- ▶ What about optimality **in the limit**?  
⇒ in general not optimal **due to inability of policy to improve**

## G3.5 Sparse Sampling

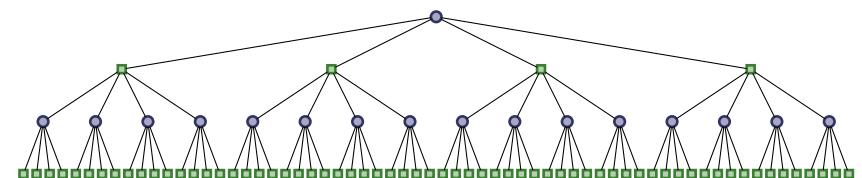
## Sparse Sampling: Idea

Sparse Sampling (Kearns et al., 2002) approaches problem that **number of reachable states under a policy** can be too large

- ▶ Creates **search tree** up to a given **lookahead horizon**
- ▶ A constant number of outcomes is **sampling** for each state-action pair
- ▶ Outcomes that were not sampled are **ignored**
- ▶ **Near-optimal**: expected cost of resulting policy close to expected cost of optimal policy
- ▶ Runtime **independent** from the number of states

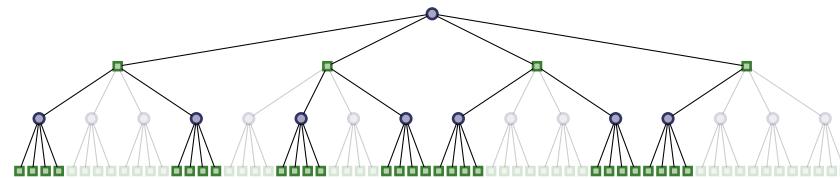
## Sparse Sampling: Search Tree

Without Sparse Sampling



## Sparse Sampling: Search Tree

With Sparse Sampling



## Sparse Sampling: Problems

- ▶ Independent from number of states, but still **exponential in lookahead horizon**
- ▶ Constants that give number of outcomes and lookahead horizon **large** for good bounds on **near-optimality**
- ▶ Search time difficult to predict
- ▶ Same amount of sampling everywhere in the tree  
⇒ resources are **wasted** in non-promising parts of the tree

## G3.6 Summary

## Summary

- ▶ Monte-Carlo methods have a long history but no successful applications until 1990s
- ▶ Monte-Carlo methods use **sampling** and **backups** that average over sample results
- ▶ **Hindsight optimization** averages over plan cost in sampled determinization
- ▶ **Policy simulation** simulates the execution of a policy
- ▶ **Sparse sampling** considers only a fixed amount of outcomes
- ▶ All three methods are **not optimal** in the limit