Planning and Optimization

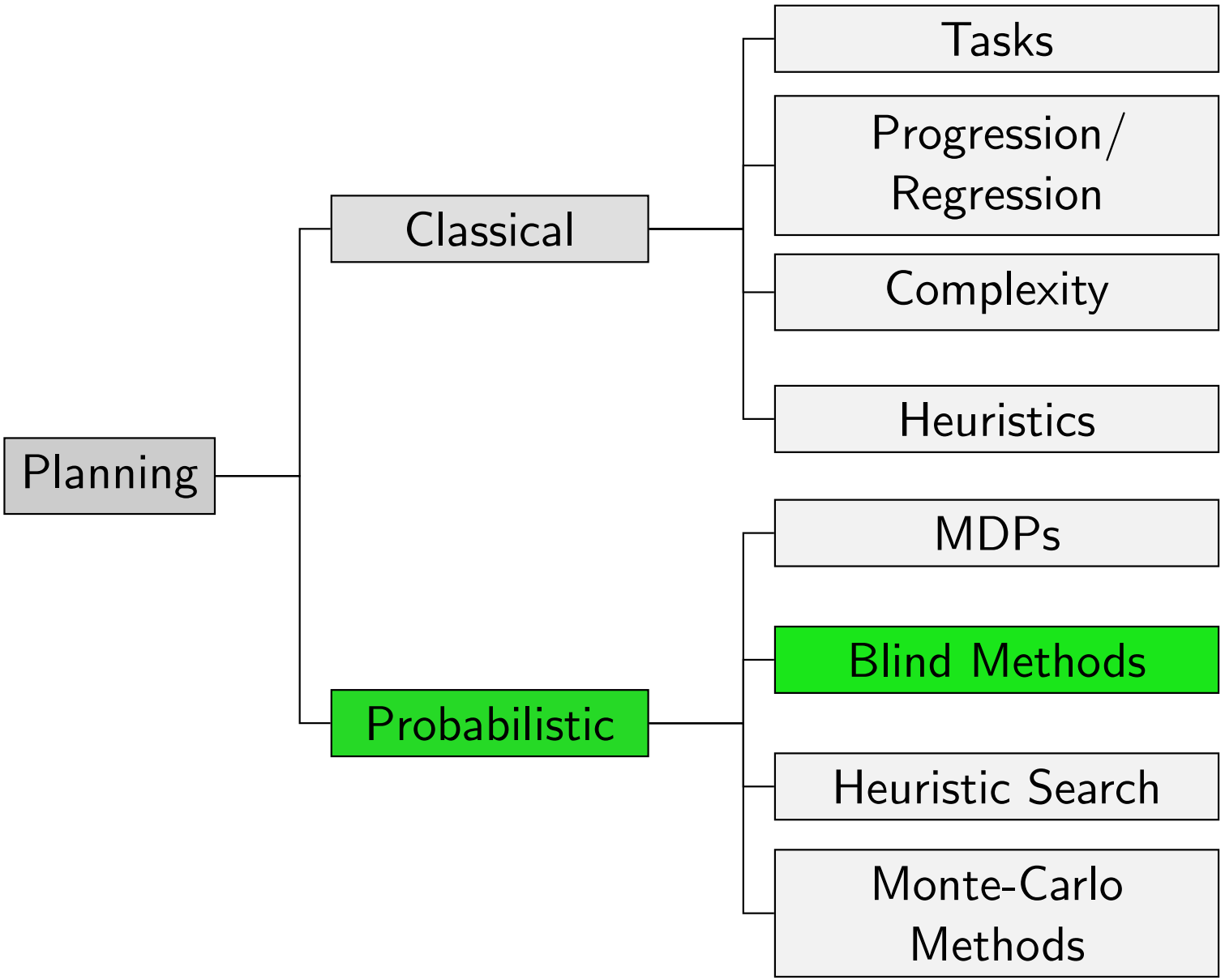
F6. Determinization-based Algorithms

Gabriele Röger and Thomas Keller

Universität Basel

November 28, 2018

Content of this Course



Determinizations in Practice

The winners of all probabilistic tracks of the International Planning Competition use determinization:

- 2004: FF-Replan (Yoon, Fern & Givan) interleaved planning & execution of plan in determinization
- 2006: FPG (Buffet & Aberdeen) learns a policy utilizing FF-Replan
- 2008: RFF (Teichteil-Königsbuch, Infantes & Kuter) extends determinization-based plan to policy
- 2011 and 2014: PROST-2011 (Keller & Eyerich) and PROST-2014 (Keller & Geißer) use determinization-based lookahead heuristic
- 2018: PROST-DD (Geißer & Speck) use BDD representation of determinization as heuristic

Determinize, Plan & Execute

Determinize, Plan & Execute: Idea

Use determinization in combination with interleaved planning & execution in determinize-plan-execute-monitor cycle for SSP $\mathcal{T}$:

- compute determinization $\mathcal{T}^d$ of $\mathcal{T}$
- use classical planner to plan action a for the current state s_0 in $\mathcal{T}^d$
- execute a
- observe new current state s'
- update $\mathcal{T}$ by setting $s_0 := s'$
- repeat until $s_0 \in S_\star$

Determinize, Plan & Execute in Practice

+ well-suited if uncertainty has certain form (e.g., actions can fail or succeed)
+ well-suited if information on probabilities noisy (e.g., path planning for robots in uncertain terrain)
+ exponential blowup through parallel probabilistic effects can be avoided (with polynomial increase of plan length)
- no technique that mitigates other weaknesses of determinizations
- gets stuck in cycle in worst case

Determinize, Plan & Execute: Implementation

- Implemented in FF-Replan (Yoon, Fern & Givan)
- uses classical planner FF (Hoffmann & Nebel)
- winner of IPC 2004
- top performer in IPC 2006, but no official competitor (used as baseline)
- led to discussions if competition domains are probabilistically interesting

Determinization Guided Policy Refinement

Determinization Guided Policy Refinement: Idea

- Plan for determinization can be seen as partial policy for all states reached by plan
- Usually not executable, as some outcomes not covered by partial policy
- Recursively plan in determinization from such an uncovered state and merge plans into policy graph
- Partial policy induced by policy graph becomes executable eventually

Determinization Guided Policy Refinement: Algorithm

1. Compute determinization $\mathcal{T}^d$ of input SSP $\mathcal{T}$ and set $s := s_0$
2. Compute plan in $\mathcal{T}^d$ from s and add all states in plan to policy graph
3. Add all uncovered outcomes to policy graph
4. Run VI on policy graph and collect all states in current solution graph without policy mapping
5. Compute probability to end up in uncovered state; terminate if smaller than some threshold
6. Choose uncovered state s' in best solution graph and set $s := s'$; repeat from 2

Determinization Guided Policy Refinement: Example

$\rightsquigarrow$ Blackboard

Determinization Guided Policy Refinement in Practice

+ optimal in the limit (if provided with unbounded deliberation time and memory)
- order in which policy graph is extended depends only on determinization and hence on plan cost (optimistic)
- while probabilities (and hence expected cost) are ignored
- weaknesses of determinizations affect early policies

Determinization Guided Policy Refinement: Implementation

- Implemented in RFF (Teichteil-Königsbuch, Infantes & Kuter)
- uses classical planner FF (Hoffmann & Nebel)
- winner of IPC 2008
- near-optimal for many benchmark problems

Lookahead in FH-MDPs

Determinization for FH-MDPs

Determinization of FH-MDP is no classical planning task:

- But: the finite horizon can be compiled into a goal:
  - add finite-domain variable v_h with $\mathrm{dom}(v_h) = \{0, \ldots, H\}$
  - $s_0(h) = H$
  - introduce $S_\star := \{s \in S \mid s(h) = 0\}$
  - add effect $s(h) := s(h) - 1$ to all operators
- However: compilation of state-dependent rewards to state-independent costs leads to exponential blowup

$\Rightarrow$ Compilation not always possible, cannot use classical planner

Lookahead Heuristic: Idea

Use determinization as heuristic:

- Search directly in determinized FH-MDP
  ($\Rightarrow$ a deterministic FH-MDP)
- Use most likely determinization for small branching factor
- To balance computation time, limit search horizon
- and use iterative deepening search that stops after time limit is reached

$\Rightarrow$ efficient lookahead in most likely future

Lookahead Heuristic in Practice

+ supports state-dependent rewards
+ balances accuracy and computation time
- probabilities (and hence expected cost) are ignored
- heuristic prone to weaknesses of determinizations
+ used only as heuristic $\Rightarrow$ search can overcome weaknesses

Lookahead Heuristic: Implementation

- Implemented in PROST-2011 (Keller & Eyerich) and PROST-2014 (Keller & Geißer)
- winner of IPC 2011 and 2014
- despite simplicity well-suited to guide search

Summary

Summary

- Winners of all probabilistic tracks of International Planning Competition use determinization
- FF-Replan uses determinize-plan-execute-monitor cycle
- RFF iteratively refines determinization-based plans to policy
- PROST uses determinization result as heuristic