Reverse Iterative Deepening for Finite-Horizon MDPs with Large Branching Factors

Andrey Kolobov, Peng Dai, Mausam, Daniel S. Weld

Computer Science and Engineering

University of Washington, Seattle

IPPC-2011

- Goal-oriented MDPs
- Small branching factors
- Optimization criterion: prob. of reaching the goal
- Solvable with:
  - Heuristic search
  - Determinization planning

- Reward MDPs, big finite horizons
- Enormous branching factors
- Optimization criterion: total expected reward
- Solvable with:
  - Heur. search? No! Much branching
  - Det. planning? No! Doesn't help

Objectives

To build a scalable planner that

- Has good anytime behavior
- Capable of dealing with FH MDPs with large branching factors and long horizons
- Generalizes beyond IPPC
  - Has few parameters

GLUTTON Overview

- Uses offline LR^{2}TDP
  - LRTDP with reverse iterative deepening
  - With some optimizations
    - Subsampling transition function
    - Correlated transition function samples
    - Caching
    - Others

UCT vs. LRTDP

UCT (Kocsis&Czepesvari, ECAI'06)

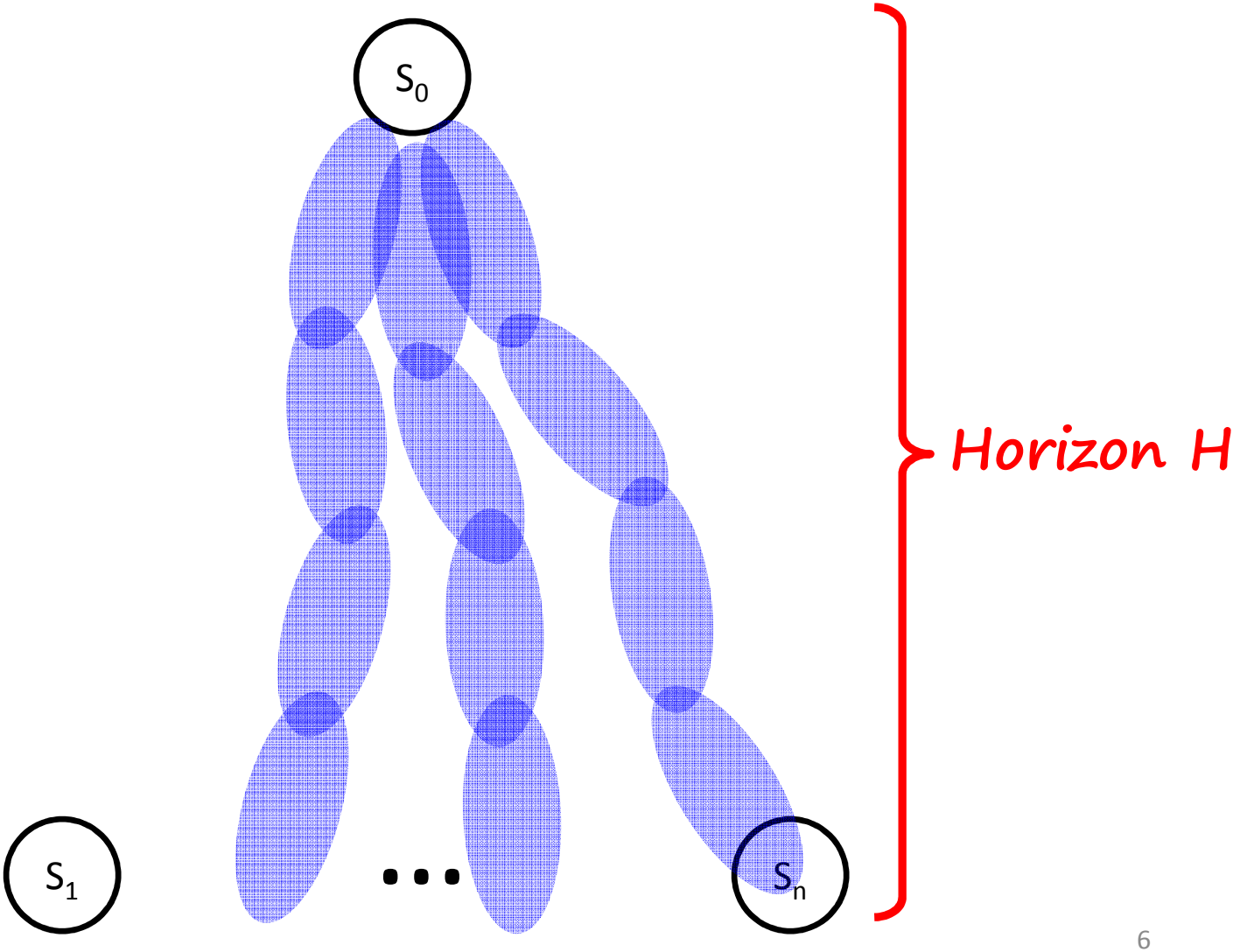
- Successful in many areas
- Handles high branching
  - If you know good params!
- Excellent anytime behavior
  - If you know good params!

LRTDP (Bonet&Geffner, ICAPS'03)

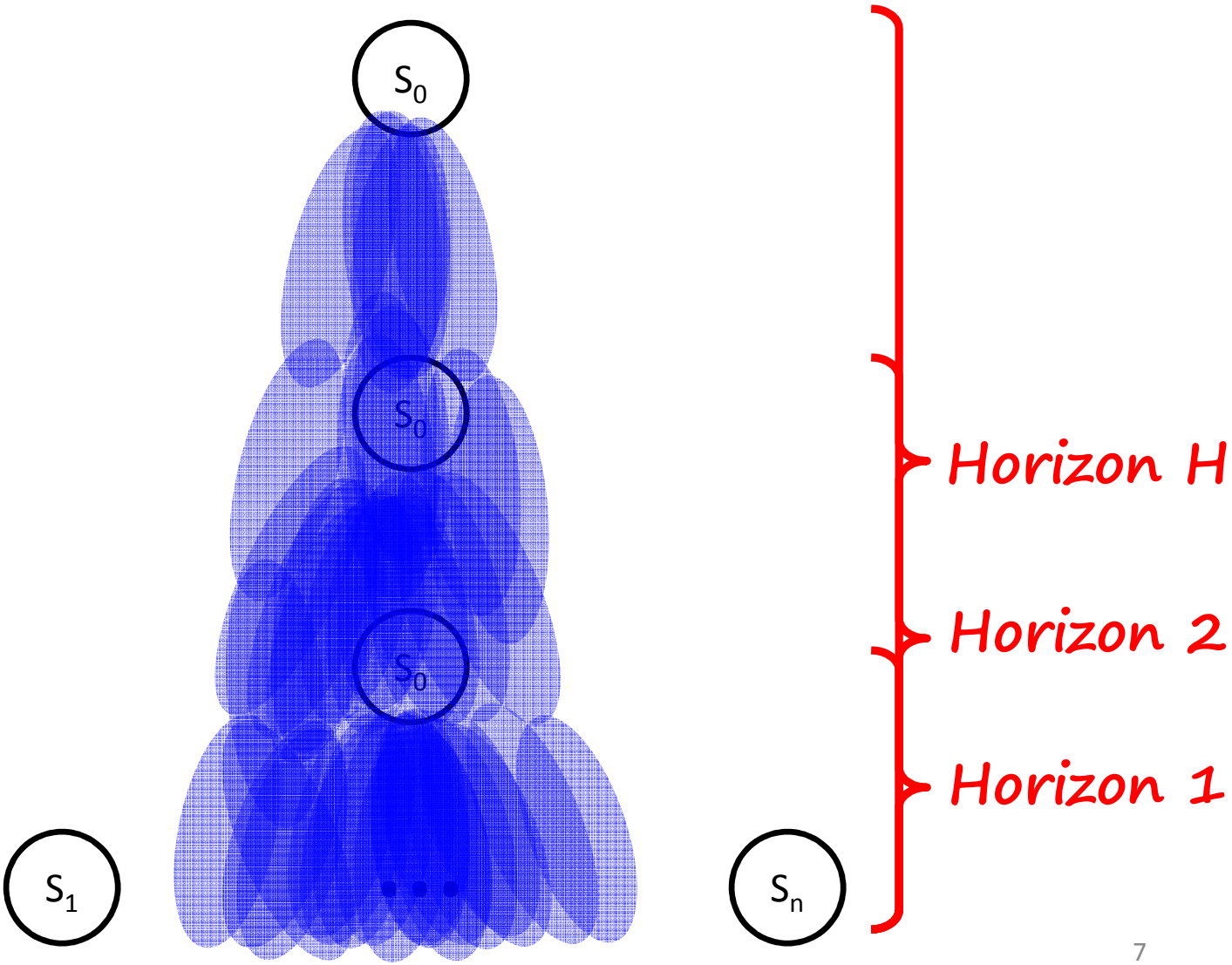
- Successful in planning
- Poor with high branching
  - Relies on Bellman backups
- Excellent anytime behavior
  - In goal-oriented problems

Hard to pick, don't generalize across problems

LRTDP in the Finite-Horizon Setting



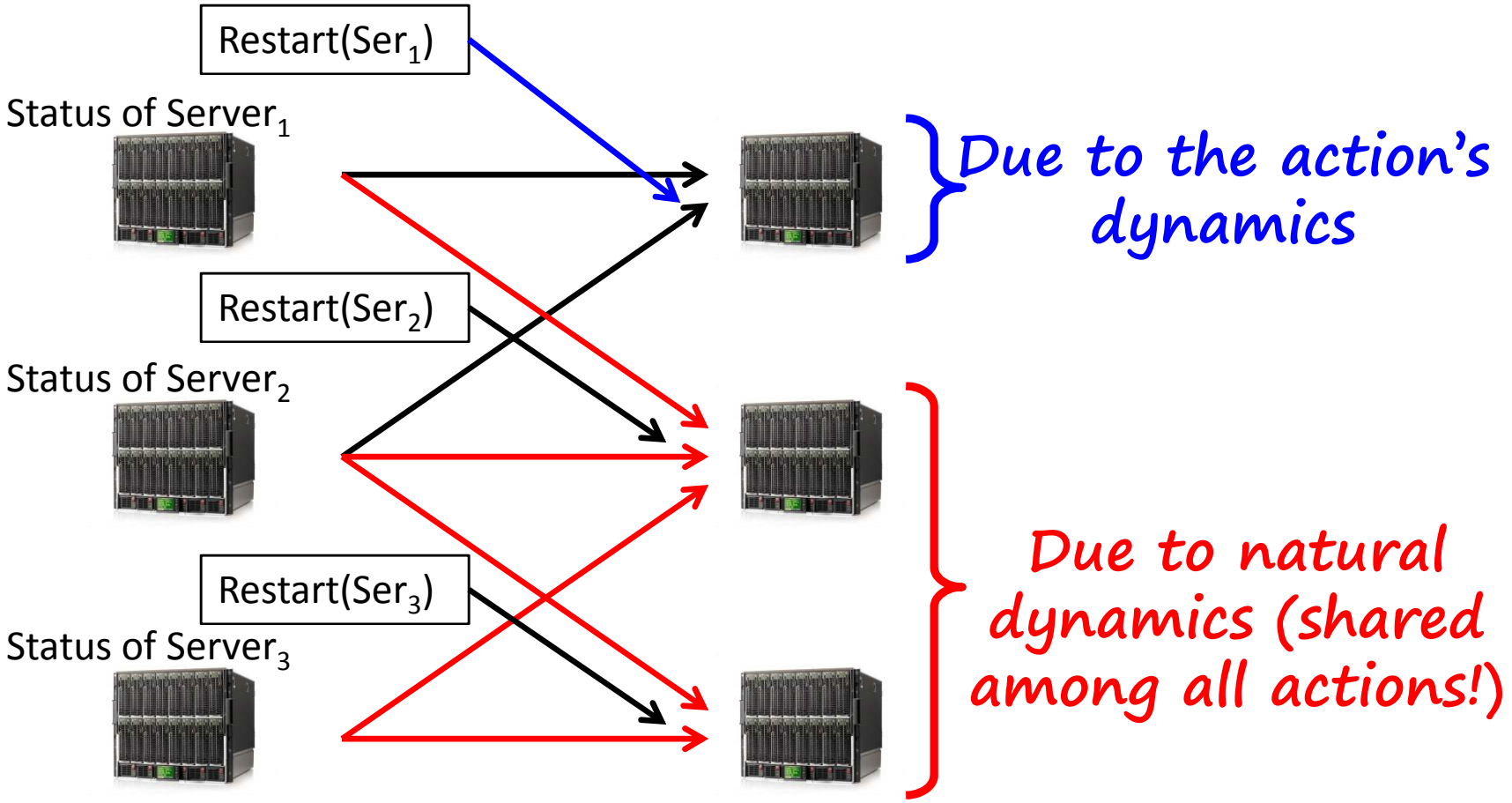
LRTDP: Reverse Iterative Deepening for Better Anytime Performance



Dealing with High Branching

- Subsample!
  - Sample several successors of *s, a*
  - Perform Bellman backups only over the samples
- Optimal as the number of samples goes to infinity

Separating Out Natural Dynamics



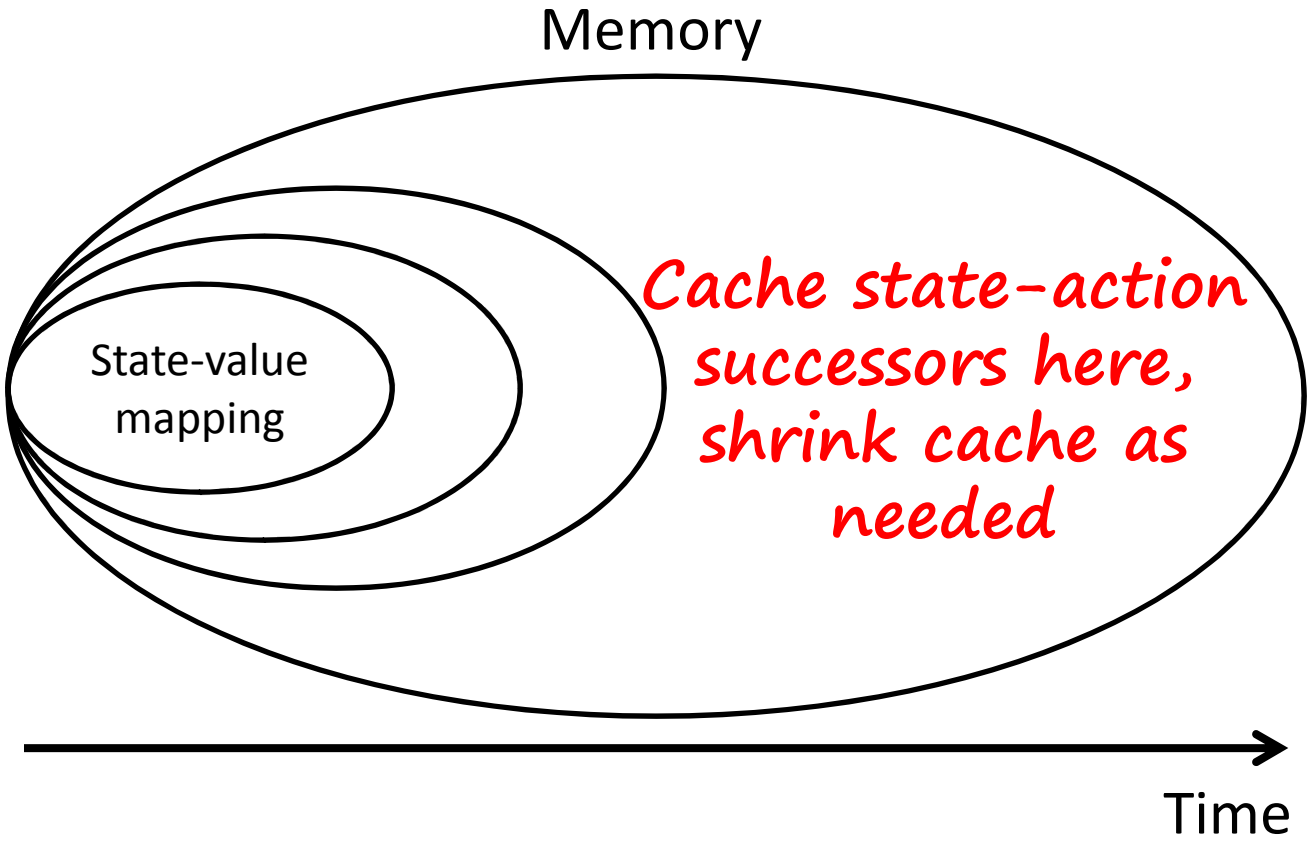
Sampling Successors of *s* under all actions: The Algorithm

- For the current state s:
  - Generate N samples of all variables affected by ND
  - For each action *a* of these N samples:
    - Resample the variables affected by AD

- Main insight: each *a* by itself affects few state vars
  - Large speedup (no need to resample ND for each *a*)
  - But… makes samples for different actions correlated

Caching State-Value Successor Samples

Observation: until planners are memory-bound, they are CPU-bound



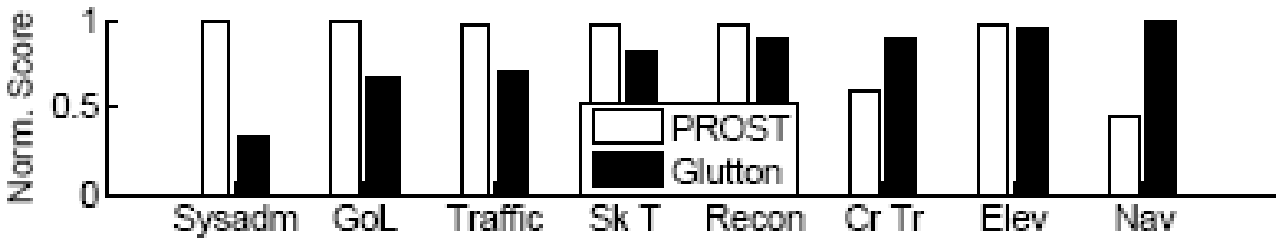
Other Optimizations

- Upper-bound heuristic
  - $H(s, t) = \max_{t' \text{ for which } s \text{ is solved}} (t-t')R_{max}(a) + V^*(s,t')$

- Default actions
  - Tell you what to do in unexplored states

Experimental Results

- LRTDP vs. PROST (Keller & Eyerich, ICAPS-2012) on all IPPC-2011 domains



- Reverse ID helps on goal-oriented domains
- Offline planning isn't worth it on large problems

Coming Up Next: Gourmand

- Same ideas as Glutton, but online
  - Given a time limit, automatically allocates time for each time step up to the horizon
  - Provides policy guarantees

- Beats both offline LR^{2}TDP (as in Glutton) and UCT (as in PROST)
  - Details as AAAI'12

Conclusions

- Presented scalable algorithm for FH MDPs with large branching factors
  - Based on offline LRTDP with reverse iterative deepening and optimizations
  - Has good anytime performance

- When used online has even better anytime performance
  - *Gourmand (Kolobov, Mausam, Weld, AAAI'12)*