Planning and Execution with Phase Transitions

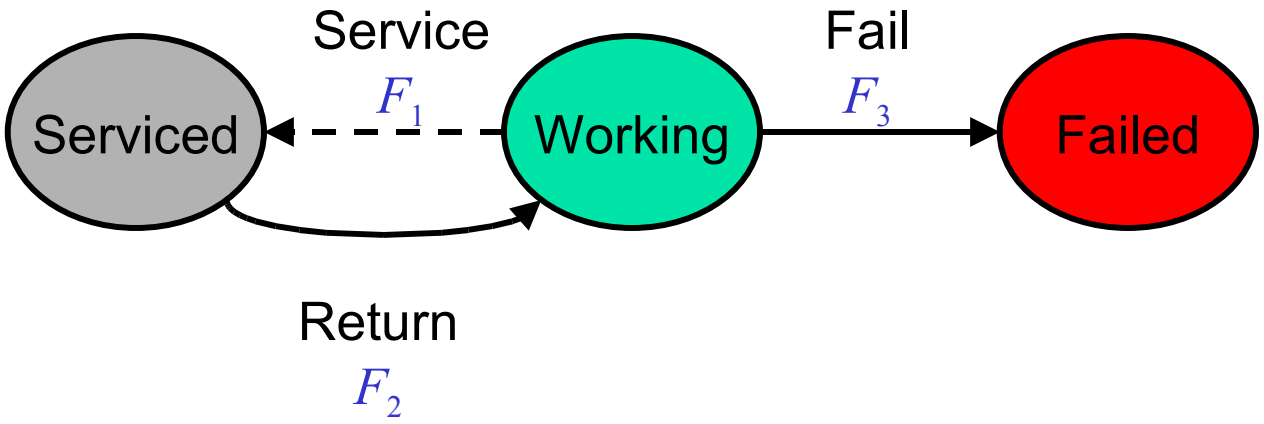
Håkan L. S. Younes

Carnegie Mellon University

Follow-up paper to Younes & Simmons'
"Solving Generalized Semi-Markov Processes using Continuous Phase-Type Distributions" (*AAAI'04*)

Planning with Time and Probability

- Temporal uncertainty
- When to schedule service?



Memoryless distributions $\Rightarrow$ MDP

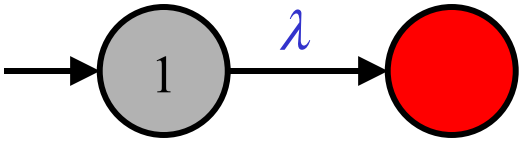
The Role of Phases

- Memoryless property of exponential distribution makes MDPs tractable
- Many phenomena are not memoryless
  - Lifetime of product or computer process
- Phases introduce memory into state space
  - Modeling tool—not part of real world (phases are hidden variables)

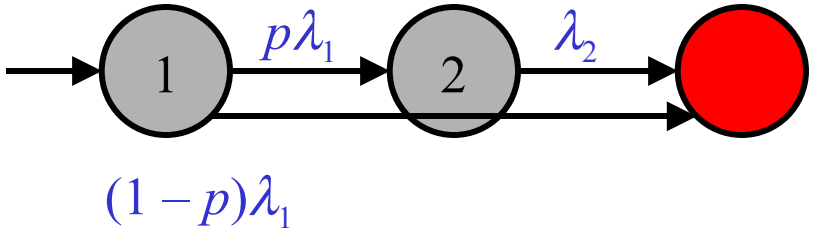
Phase-Type Distributions

- Time to absorption in Markov chain with n transient states and one absorbing state

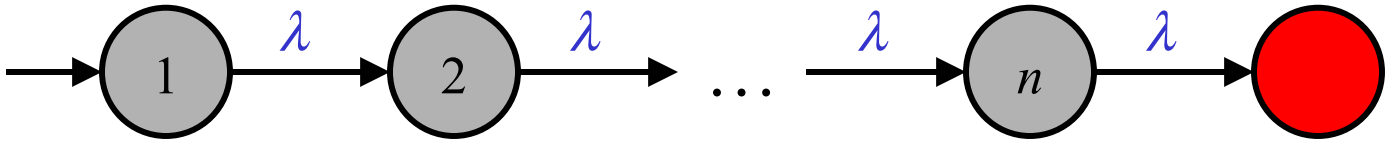
Exponential



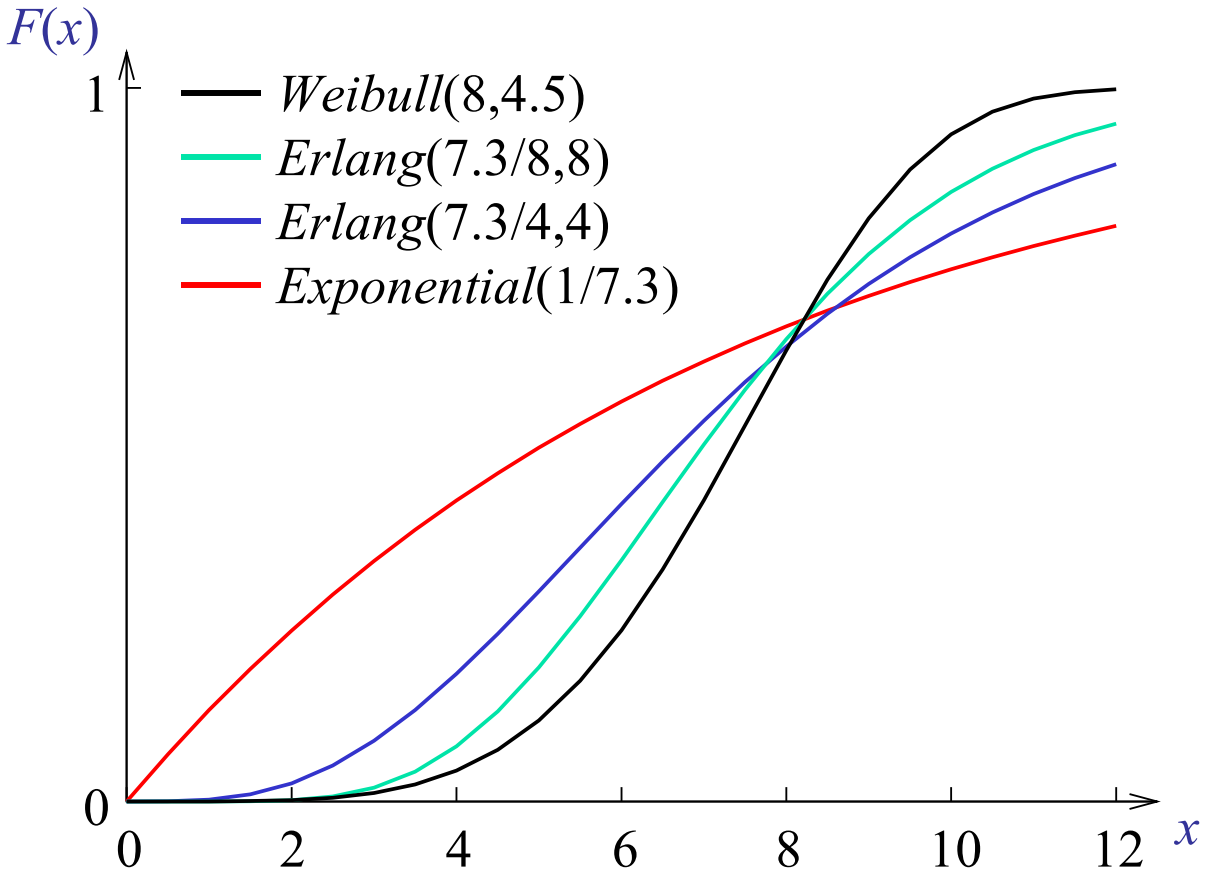
Two-phase Coxian



n-phase Erlang



Phase-Type Approximation



Solving MDPs with Phase Transitions

- Solve MDP as if phases were observable
- Maintain belief distribution over phase configurations during execution
  - AAAI'04 paper: Simulate phase transitions
- Use Q_{MDP} value method to select actions
  - OK, because there are typically no actions that are useful for observing phases

Factored Event Model

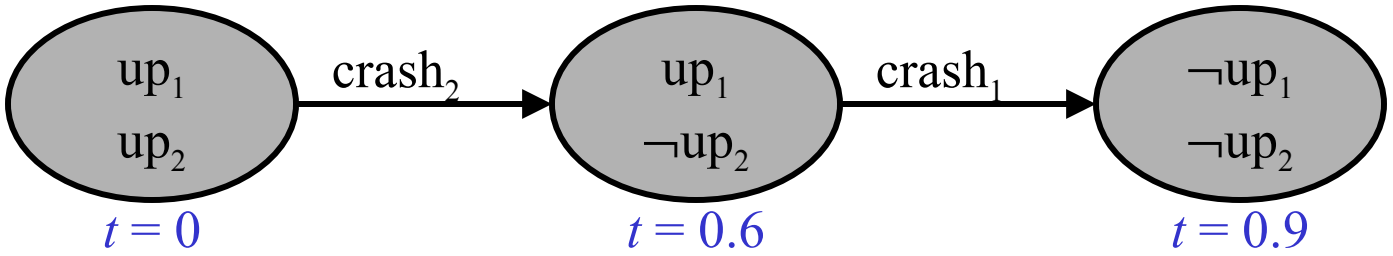
- Asynchronous events and actions
- Each event/action has:
  - Enabling condition ϕ_e
  - (Phase-type) distribution G_e governing the time e must remain enabled before it triggers
  - A distribution $p_e(s'; s)$ determining the probability that the next state is s' if e triggers in state s

Event Example

- Computer cluster with crash event and reboot action for each computer
- Crash event for computer i:
  - $\phi_i = \text{up}_i$; $p(s'; s) = 1$ iff $s \models \text{up}_i$ and $s' \models \neg\text{up}_i$
- Reboot action for computer i:
  - $\phi_i = \neg\text{up}_i$; $p(s'; s) = 1$ iff $s \models \neg\text{up}_i$ and $s' \models \text{up}_i$

Event Example

- Cluster with two computers



Exploiting Structure

- Value iteration with ADDs:

$$\vec{Q}(a) = \vec{R}_a^{\alpha} + \mathbf{P}_a^{\alpha} \cdot \vec{V}^*$$

- Factored state representation means that some variable assignments may be invalid
  - Phase is one for disabled events
  - Binary encoding of phases with $n \neq 2^k$
- Value iteration with state filter f:

$$\vec{Q}(a) = f \circ \left(\vec{R}_a^{\alpha} + \mathbf{P}_a^{\alpha} \cdot \vec{V}^* \right)$$

Phase Tracking

- Infer phase configurations from observable features of the environment
  - Physical state of process
  - Current time
- Transient analysis for Markov chains
  - The probability of being in state s at time t

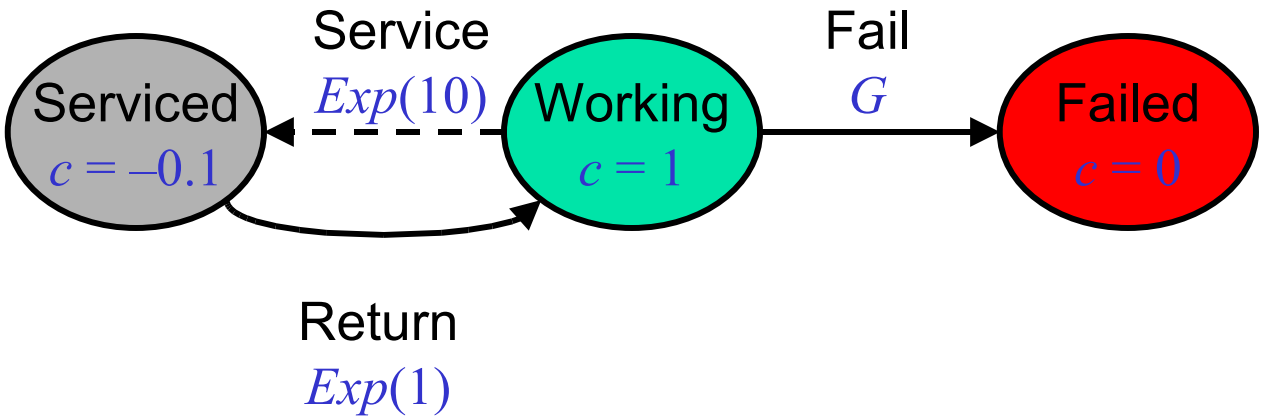
$$\vec{p}(s,t) = \vec{\pi} e^{\mathbf{Q}t}$$

Phase Tracking

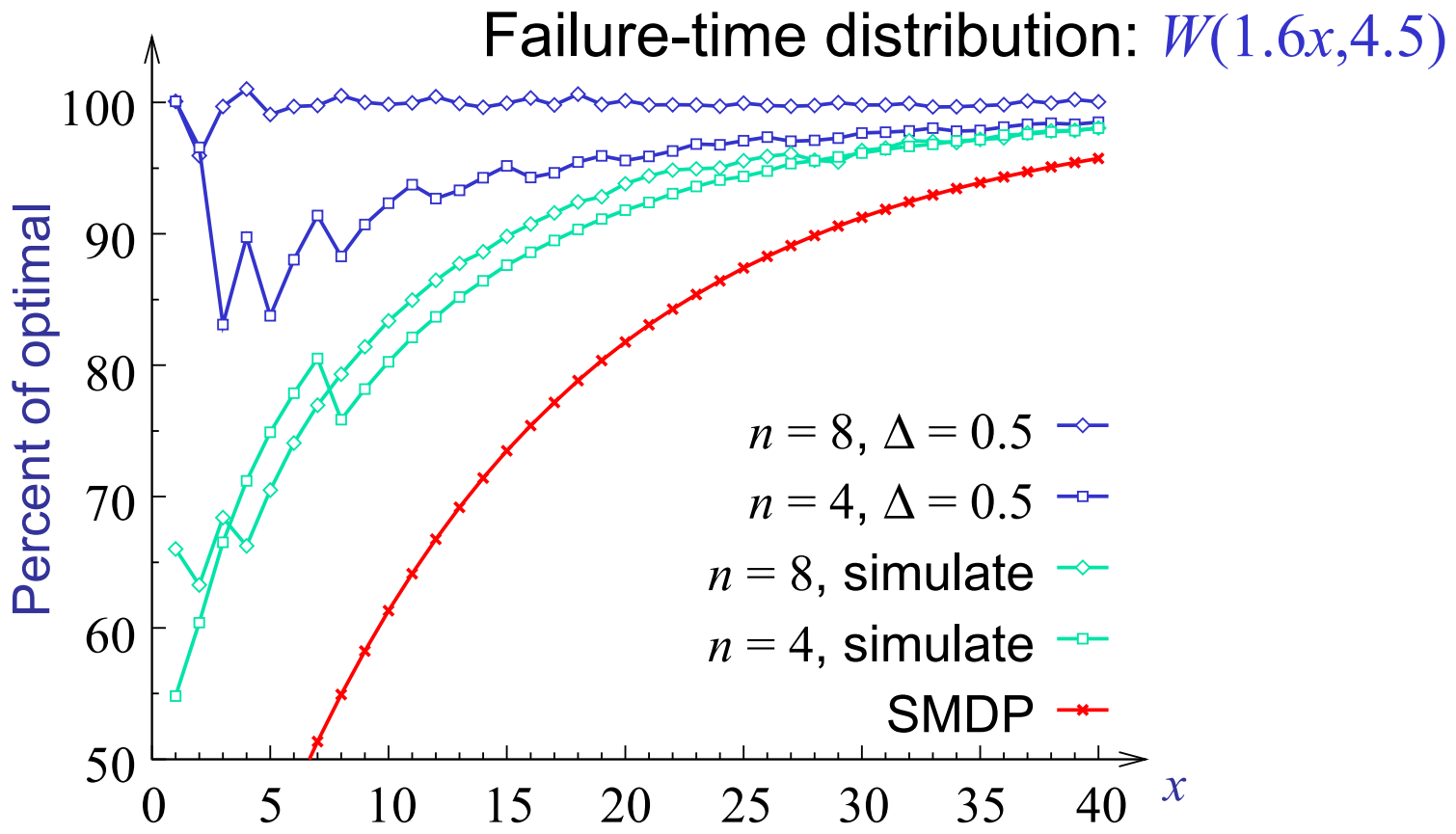
- Belief distribution changes continuously
  - Use fixed update interval Δ
- Independent phase tracking for each event
  - $\mathbf{Q}$ matrix is $n \times n$ for n-phase distribution
  - Phase tracking is $O(n)$ for Erlang distribution

The Foreman's Dilemma

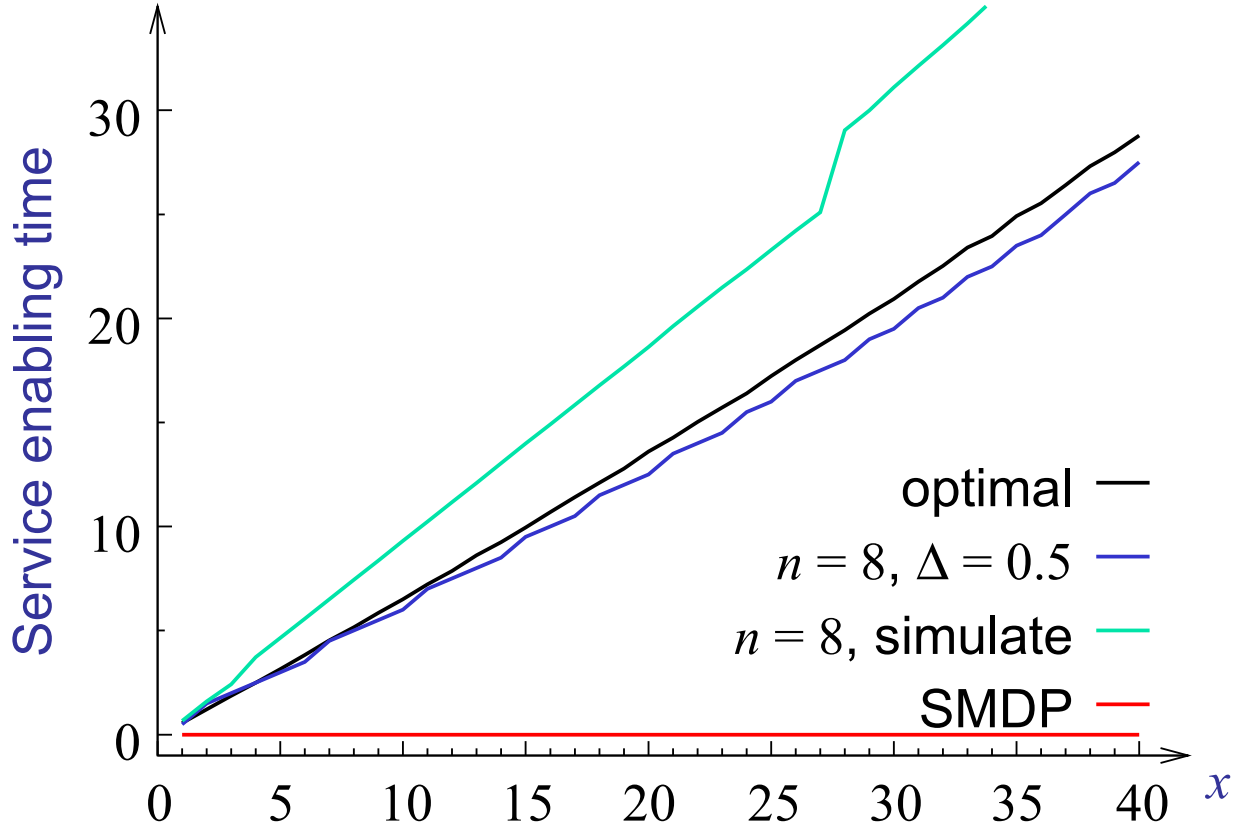
- When to enable "Service" action in "Working" state?



Foreman's Dilemma: Policy Performance



Foreman's Dilemma: Enabling Time for Action

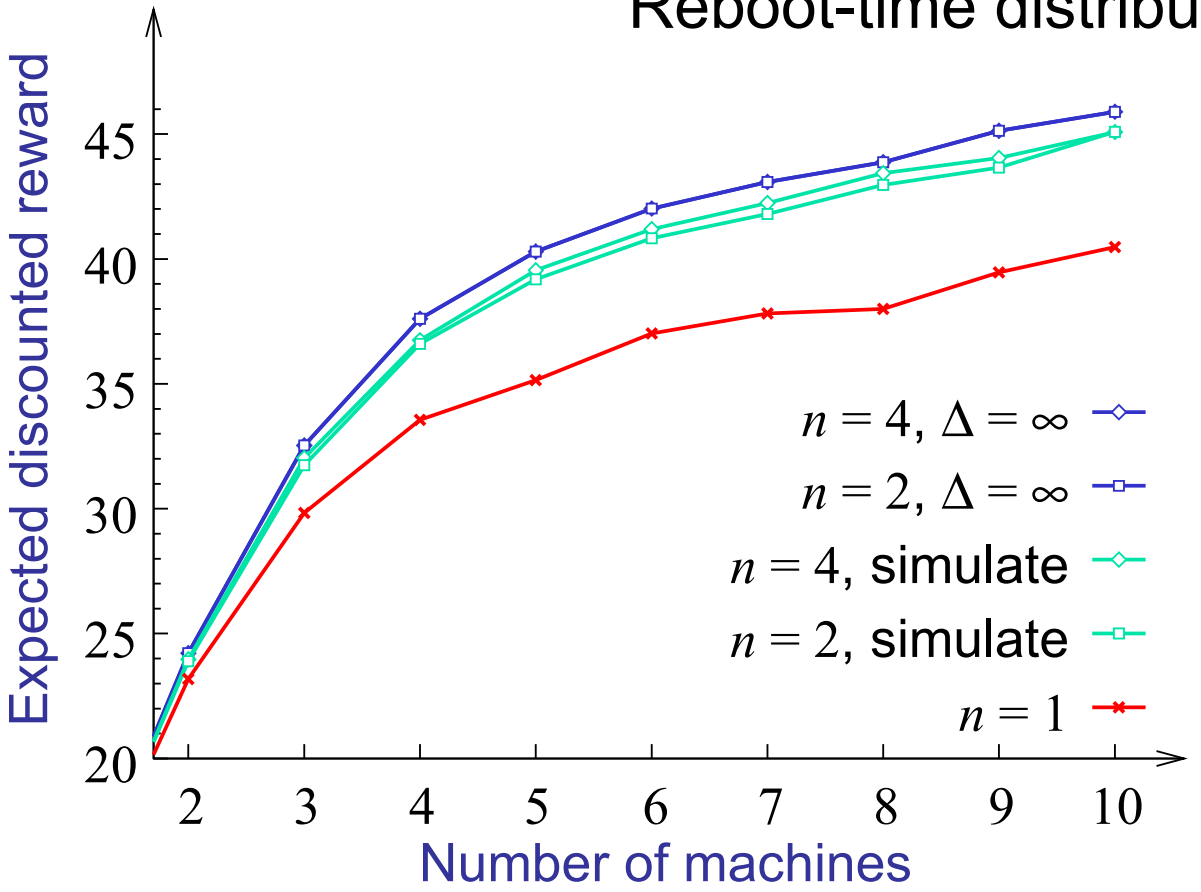


System Administration

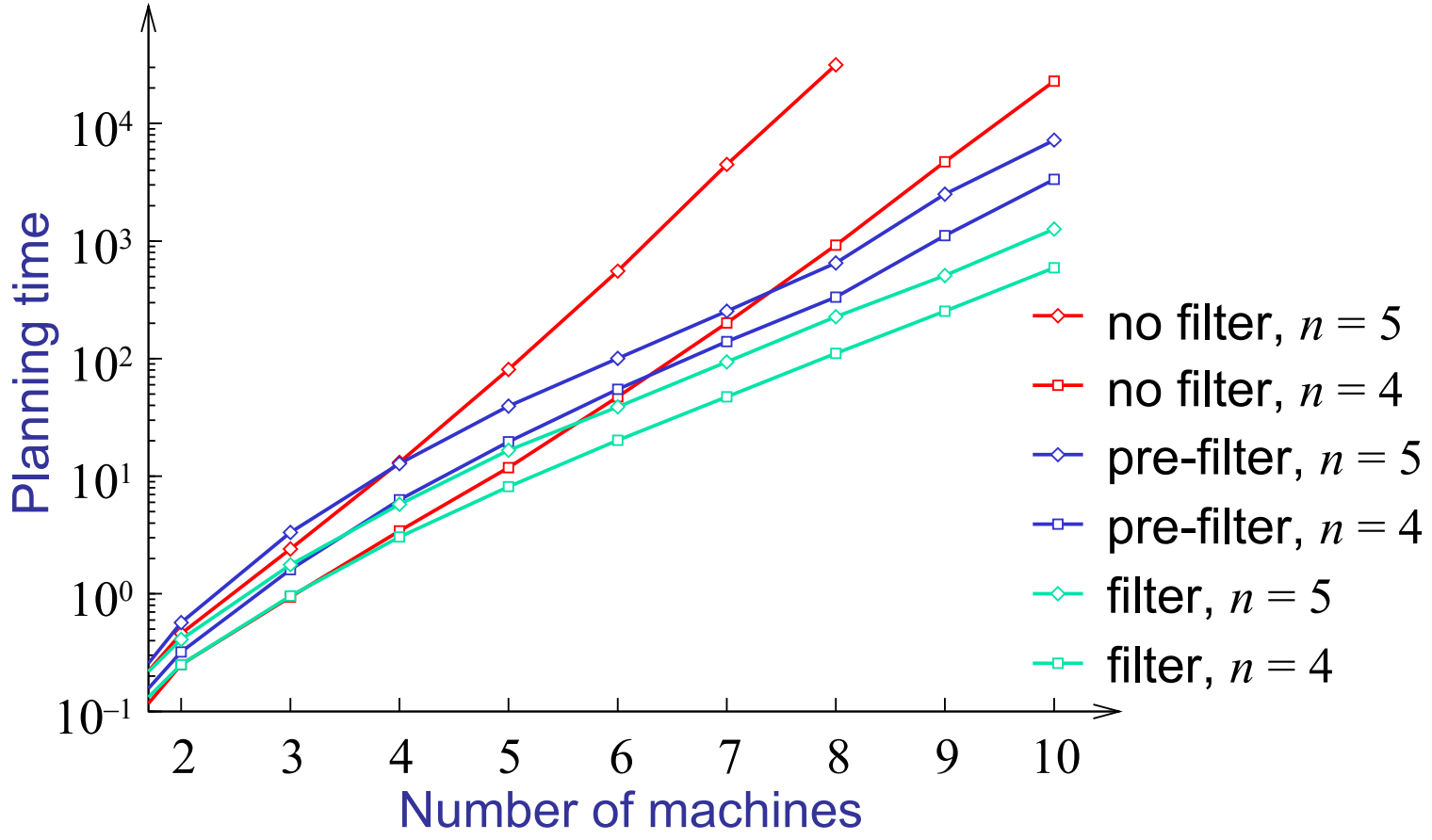
- Network of n machines
- Reward rate $c(s) = k$ in states where k machines are up
- One crash event and one reboot action per machine
  - At most one action enabled at any time (single agent)

System Administration: Policy Performance

Reboot-time distribution: $U(0,1)$



System Administration: Planning Time



Discussion

- Phase-type approximations add state variables, but structure can be exploited
  - Use approximate solution methods (e.g., Guestrin et al., *JAIR* **19**)
- Improved phase tracking using transient analysis instead of phase simulation
- Recent CTBN work with phase-type dist.:
  - modeling—not planning

Tempastic-DTP

- A tool for GSMDP planning:

  http://sweden.autonomy.ri.cmu.edu/tempastic/