Verification and Planning for Stochastic Processes with Asynchronous Events

Håkan L. S. Younes

Carnegie Mellon University

Thesis Committee:
Reid Simmons, Chair
Edmund Clarke
Geoffrey Gordon
Jeff Schneider
David Musliner, Honeywell Laboratories

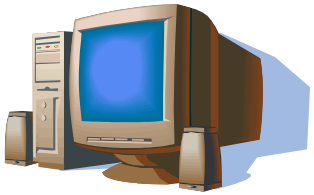
Introduction

- Asynchronous processes are abundant in the real world
  - Telephone system, computer network, etc.
- Randomness due to uncertainty in timing of events
  - For example, duration of phone call (timing of "hang up" event)

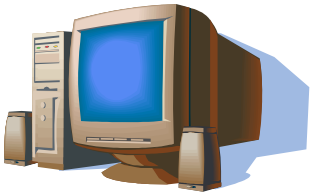
Two Problems

- Verification
  - Given an asynchronous system (or model thereof), test whether some property holds
- Planning
  - In the presence of asynchronous events, find policy that satisfies specified objectives

Illustrative Example: System Administration



m_1

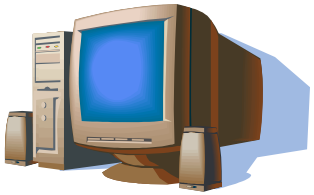


m_2

m_1 up
m_2 up

t = 0

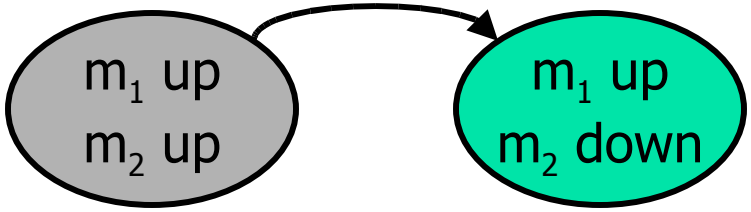
Illustrative Example: System Administration



m_1

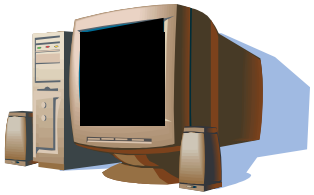
m_2

m_2 crashes



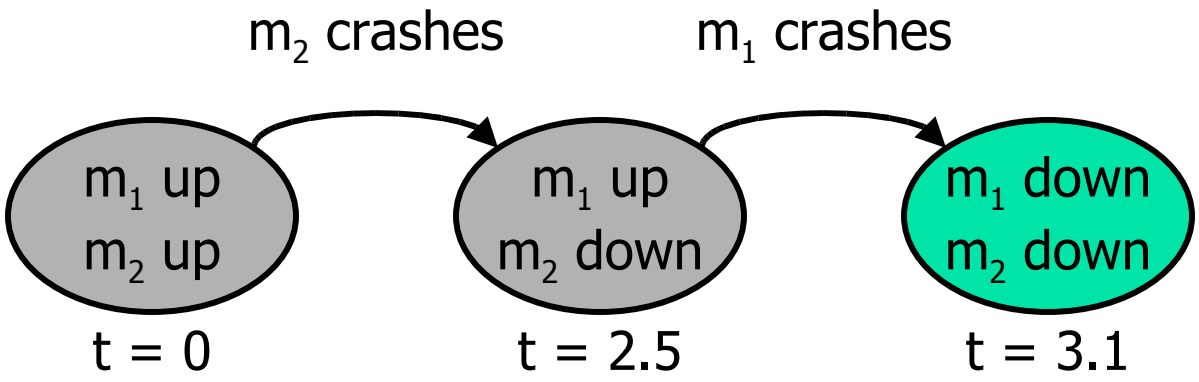
t = 0 t = 2.5

Illustrative Example: System Administration

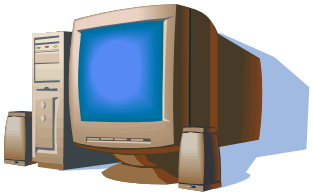


m_1

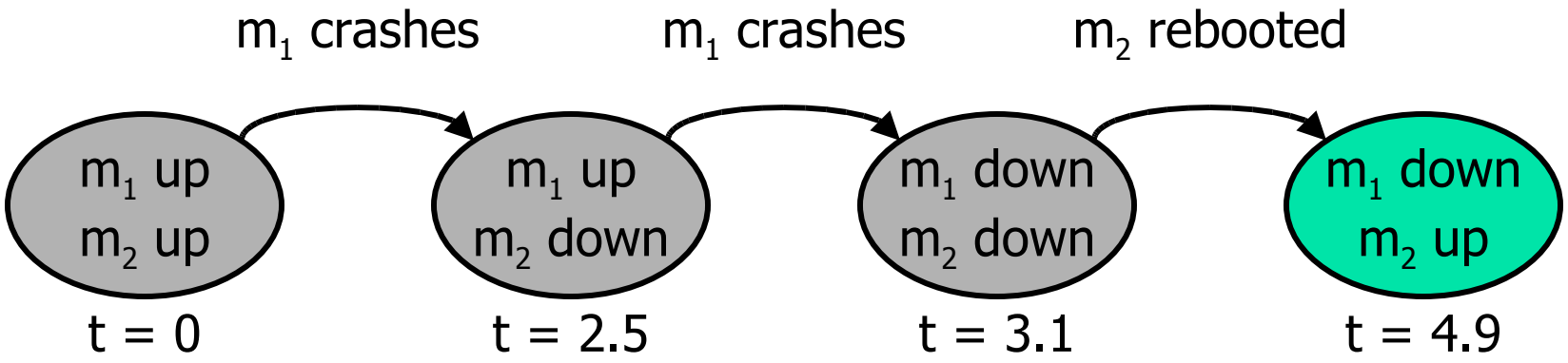
m_2



Illustrative Example: System Administration



Stochastic discrete event system

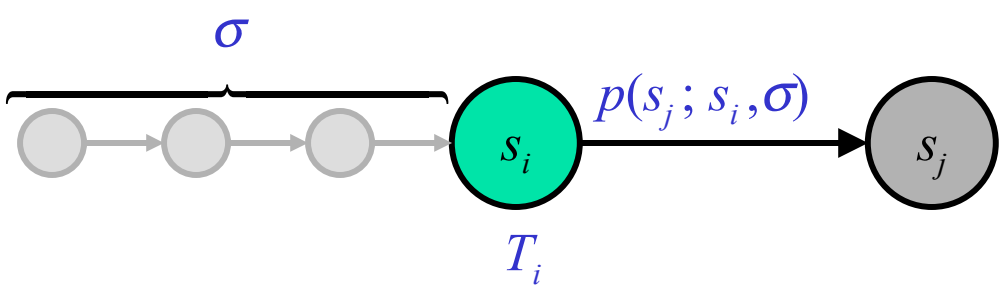


Illustrative Example: System Administration

- Verification
  - Test whether the probability is at most 0.1 that both machines are simultaneously down within the first hour of operation
- Planning
  - Find a service policy that maximizes uptime

Stochastic Discrete Event Systems

- State transitions are caused by events
  - State holding time is a random variable
  - Probability distribution over next states



$$\Pr[T_i \le t] = F(t\,;\,\sigma)$$

Why Challenging?

- History dependence
  - State holding time distributions may not be memoryless
  - Continuous state variables may be required to encode execution history in state space
- Continuous-time
  - Asynchronous events may appear simultaneous with any time discretization

Thesis

"Verification and planning for stochastic processes with asynchronous events can be made practical through the use of statistical hypothesis testing and phase-type distributions"

Summary of Contribution: Verification

- Unified logic for transient properties of stochastic discrete event systems
- Statistical probabilistic model checking
  - Statistical hypothesis testing and discrete event simulation
  - Conjunctive and nested probabilistic formulae
  - Distributed sampling
- Statistical verification of “black-box” systems

Summary of Contribution: Planning

- Framework for stochastic decision processes with asynchronous events
- Goal directed planning
  - Policy generation using deterministic planner
  - Policy debugging using sample path analysis
- Decision theoretic planning
  - Generalized semi-Markov decision process
  - Approximate solution using Phase-type distributions

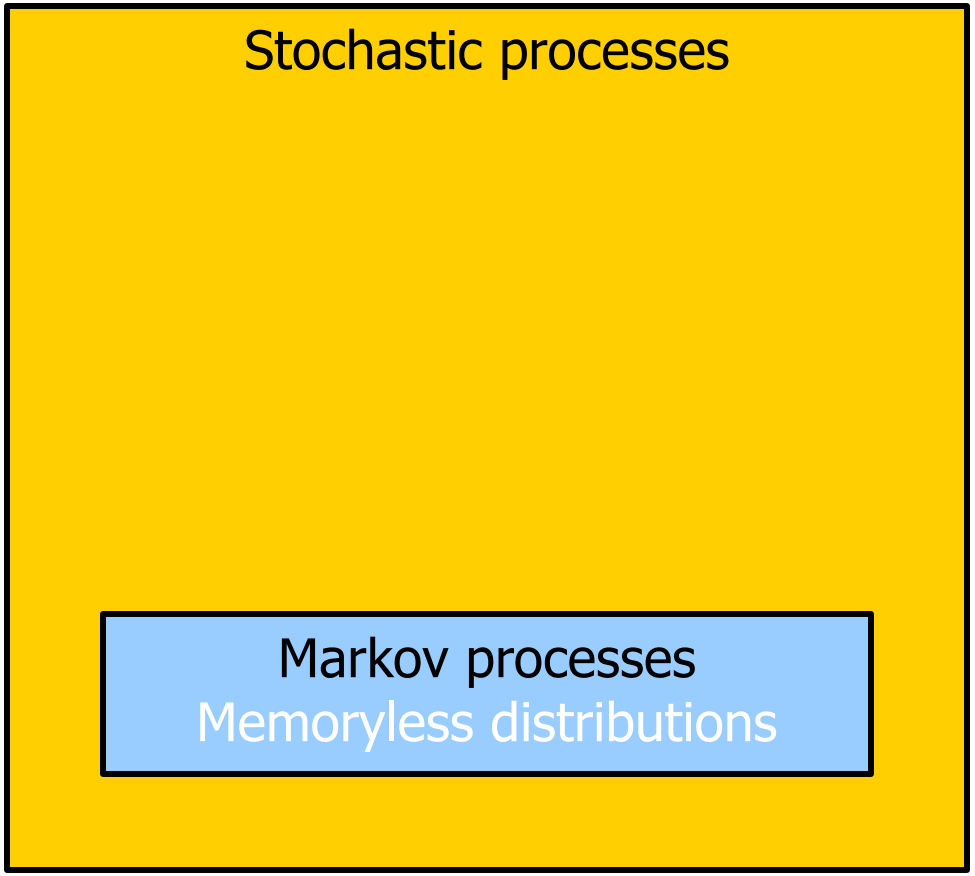
Relation to Previous Research

Verification

Discrete time
[Hansson & Jonsson 1994]
Continuous time
[Baier et al. 2003]

Planning

Markov decision process (MDP)
[Bellman 1957; Howard 1960]
Concurrency
[Guestrin et al. 2002;
Mausam & Weld 2004]
Value function approximation
[Bellman et al. 1963; Gordon 1995;
Guestrin et al. 2003]



Relation to Previous Research

Verification

[Infante López et al. 2001]

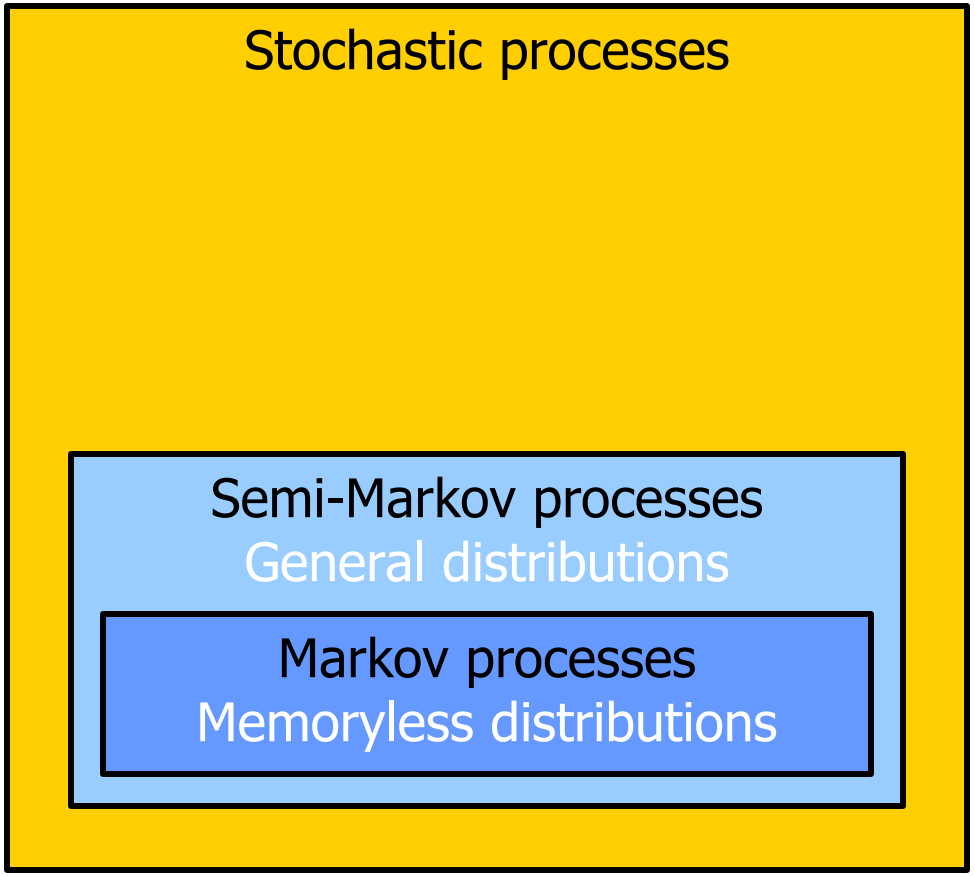
Planning

Semi-MDP

[Howard 1963]

Time dependent policies

[Chitgopekar 1969; Stone 1973; Cantaluppi 1984]



Relation to Previous Research

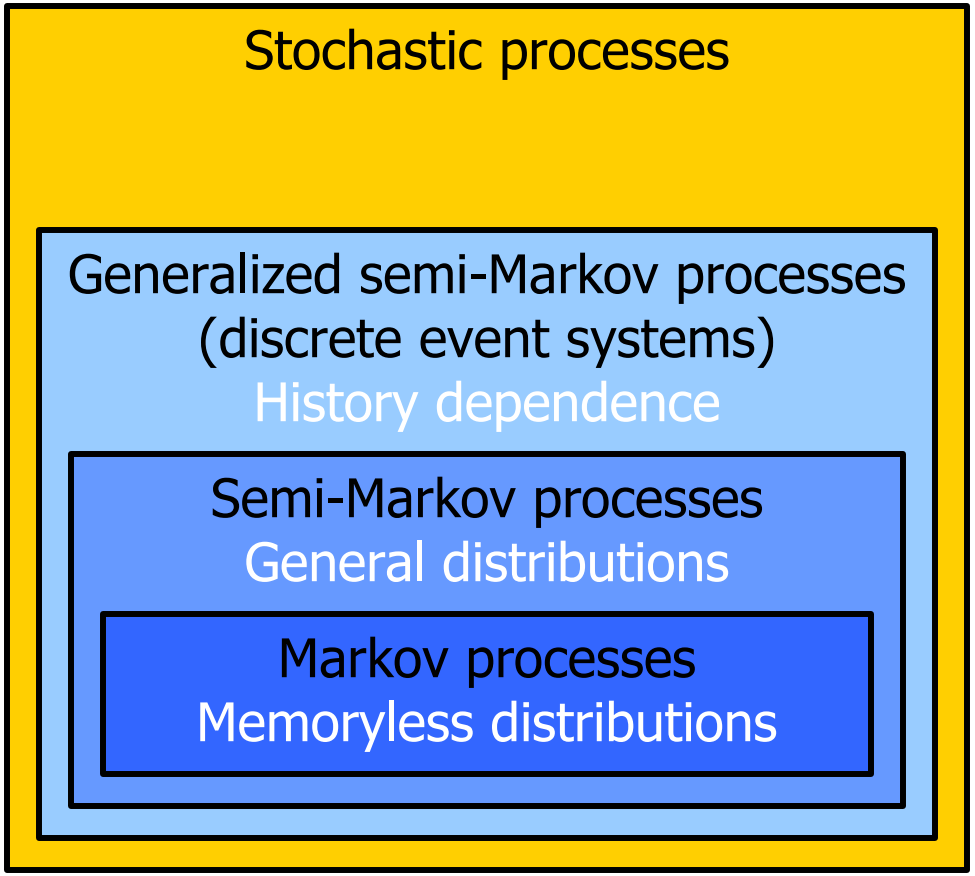
Verification

"Qualitative" properties
[Alur et al. 1991]
Probabilistic timed automata
[Kwiatkowska et al. 2000]

Planning

CIRCA (no probabilities)
[Musliner et al. 1995]
Probabilistic CIRCA
[Atkins et al. 1996]

Scope of this thesis



Topics for Remainder of Presentation

- Statistical probabilistic model checking
  - Unified Temporal Stochastic Logic (UTSL)
  - Acceptance sampling
  - Nested probabilistic statements
- Decision theoretic planning
  - Generalized semi-Markov decision processes (GSMDPs)
  - Phase-type distributions

Probabilistic Model Checking

- Given a model M, a state s, and a property Φ, does Φ hold in s for M?
  - Model: stochastic discrete event system
  - Property: probabilistic temporal logic formula

Unified Temporal Stochastic Logic (UTSL)

- Standard logic operators: $\neg\Phi$, $\Phi \wedge \Psi$, …
- Probabilistic operator: $\mathcal{P}_{\geq\theta}[\varphi]$
  - Holds in state s iff probability is at least θ for paths satisfying φ and starting in s
- Until: $\Phi\, \mathcal{U}^{\leq T}\, \Psi$
  - Holds over path σ iff Ψ becomes true along σ before time T, and Φ is true up to that point in time

UTSL Example

- Probability is at most 0.1 that two machines are simultaneously down within the first hour of operation
  - $\mathcal{P}_{\leq 0.1}[\top\, \mathcal{U}^{\leq 60}\, down{=}2]$

Statistical Solution Method

- Use discrete event simulation to generate sample paths
- Use acceptance sampling to verify probabilistic properties
  - Hypothesis: $\mathcal{P}_{\geq\theta}[\varphi]$
  - Observation: verify φ over a sample path

Not estimation!

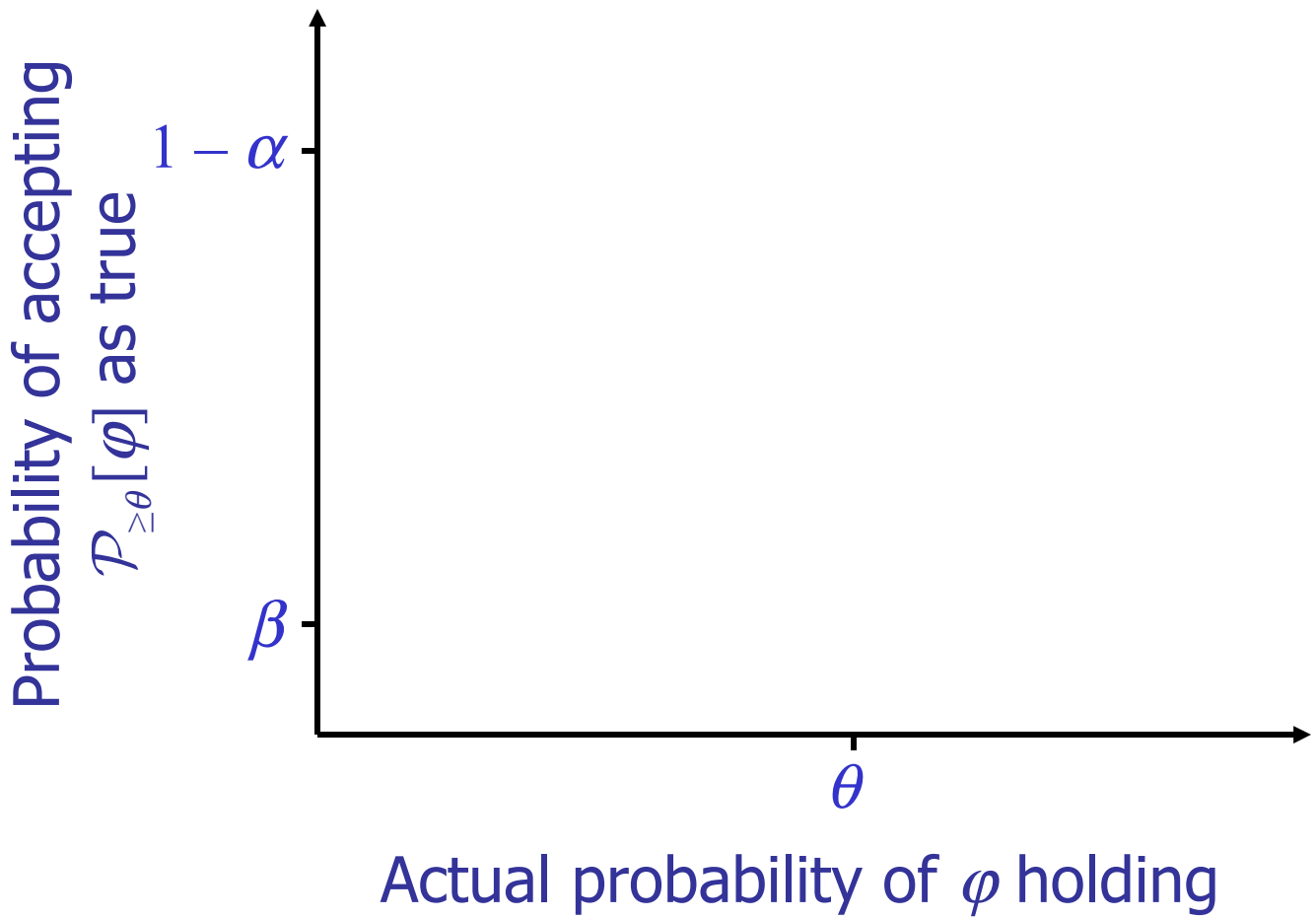
Why Statistical Approach?

- Benefits
  - Insensitive to size of system
  - Easy to trade accuracy for speed
  - Easy to parallelize
- Alternative: Numerical approach
  - High accuracy in result
  - Memory intensive
  - Limited to certain classes of systems

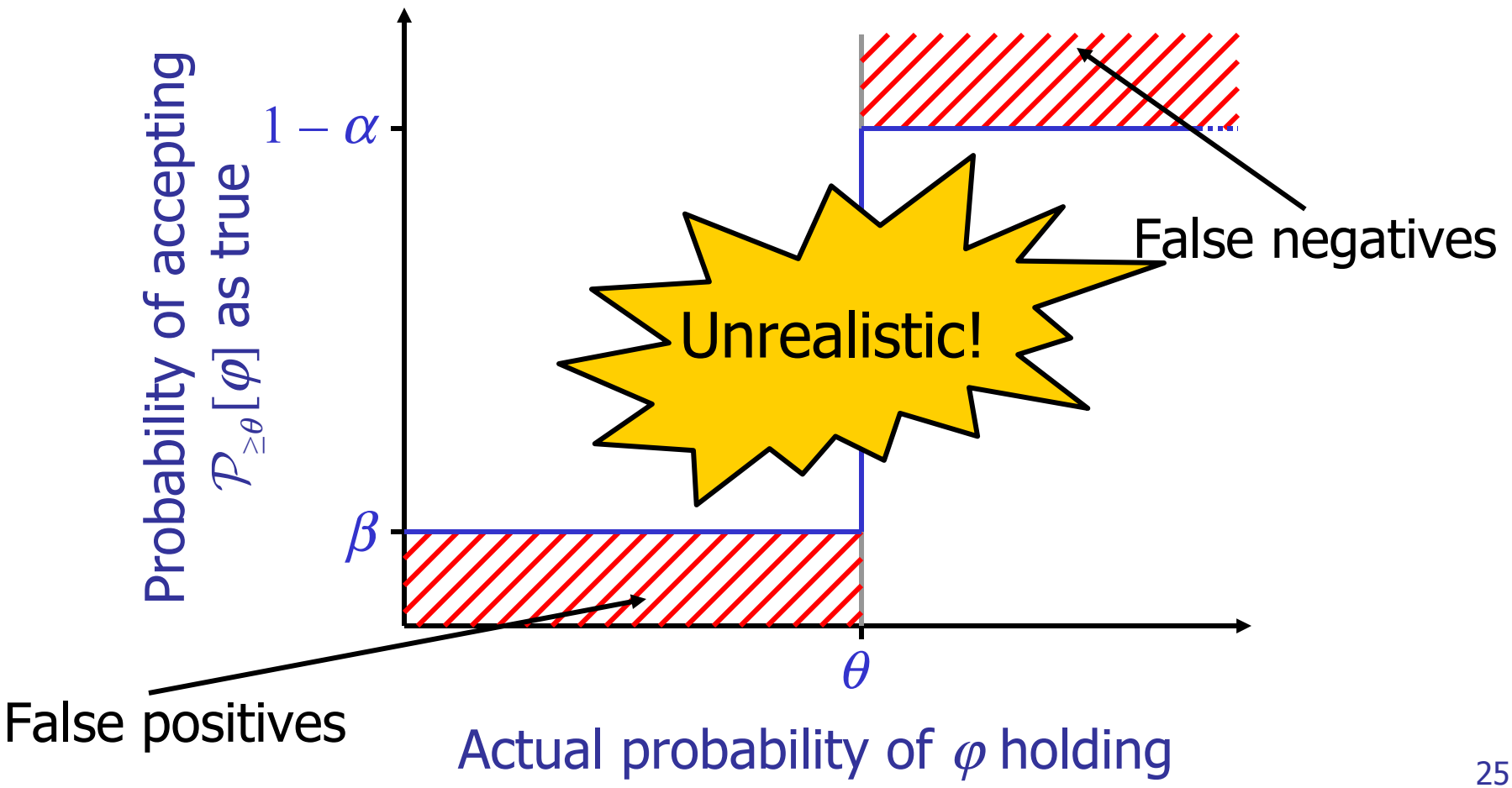
Error Bounds

- Probability of false negative: $\leq \alpha$
  - We say that Φ is false when it is true
- Probability of false positive: $\leq \beta$
  - We say that Φ is true when it is false

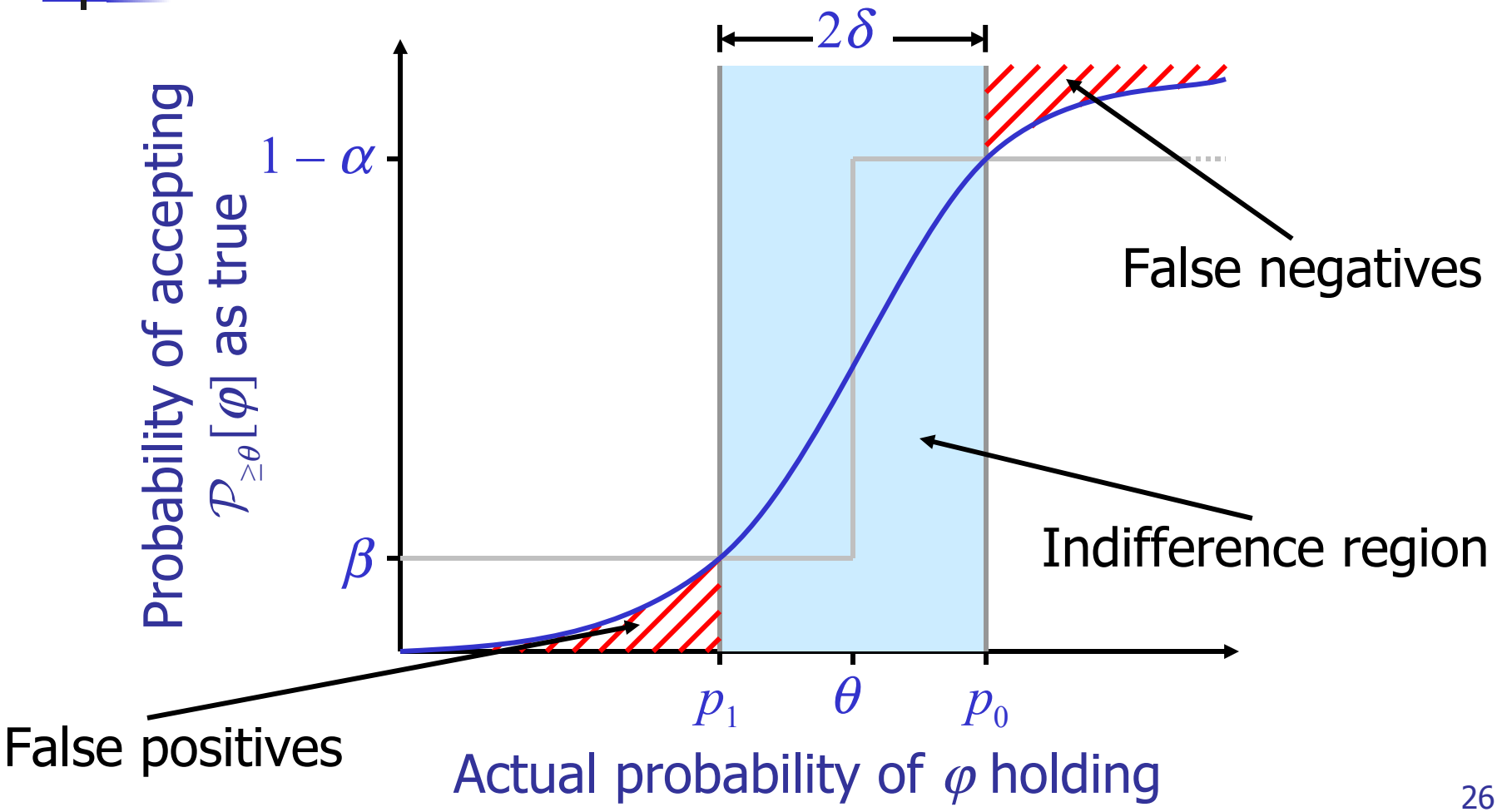
Performance of Test



Ideal Performance of Test



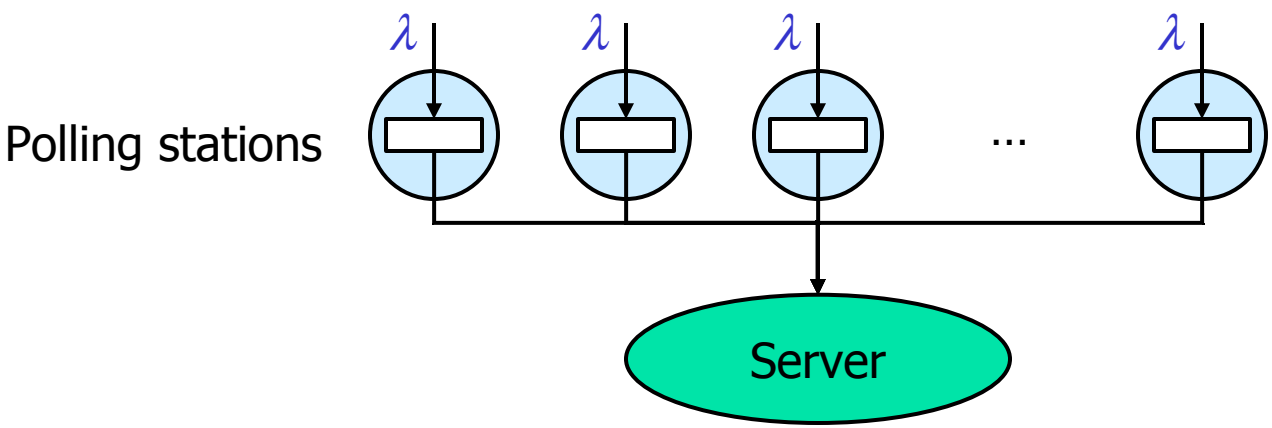
Realistic Performance of Test



Sequential Acceptance Sampling [Wald 1945]

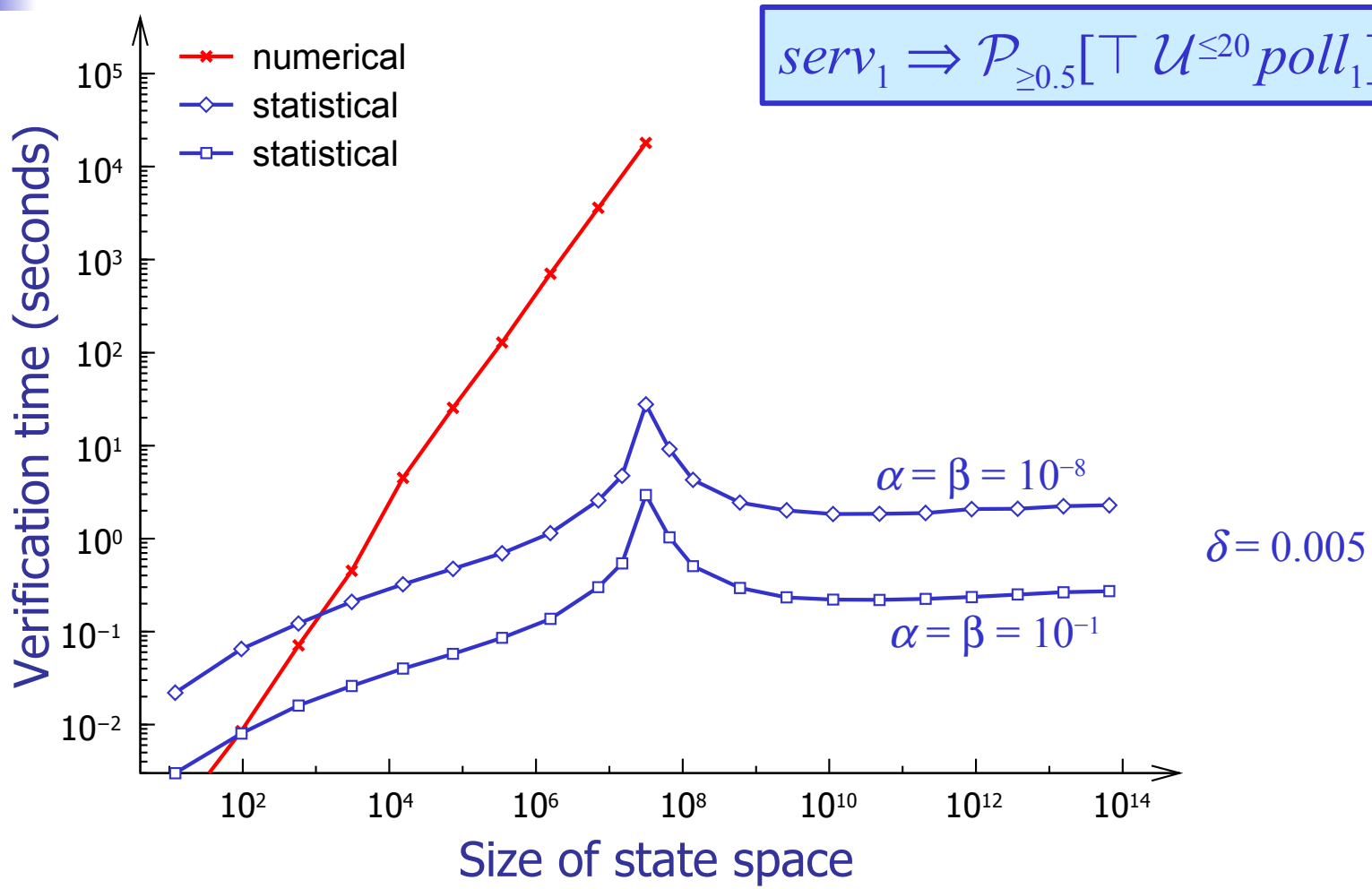
Case Study: Symmetric Polling System

- Single server, n polling stations
- Stations are attended in cyclic order
- Each station can hold one message
- State space of size $O(n \cdot 2^n)$



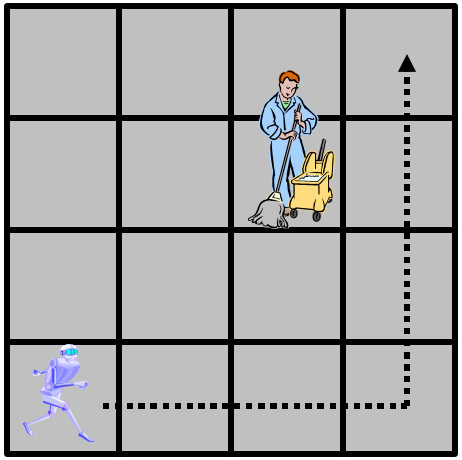
Symmetric Polling System (results)

$$serv_1 \Rightarrow \mathcal{P}_{\geq 0.5}[\top\, \mathcal{U}^{\leq 20}\, poll_1]$$



Nested Probabilistic Statements: Robot Grid World

- Probability is at least 0.9 that goal is reached within 100 seconds while periodically communicating
  - $\mathcal{P}_{\geq 0.9}[\mathcal{P}_{\geq 0.5}[\top\ \mathcal{U}^{\leq 9}\ comm]\ \mathcal{U}^{\leq 100}\ goal]$

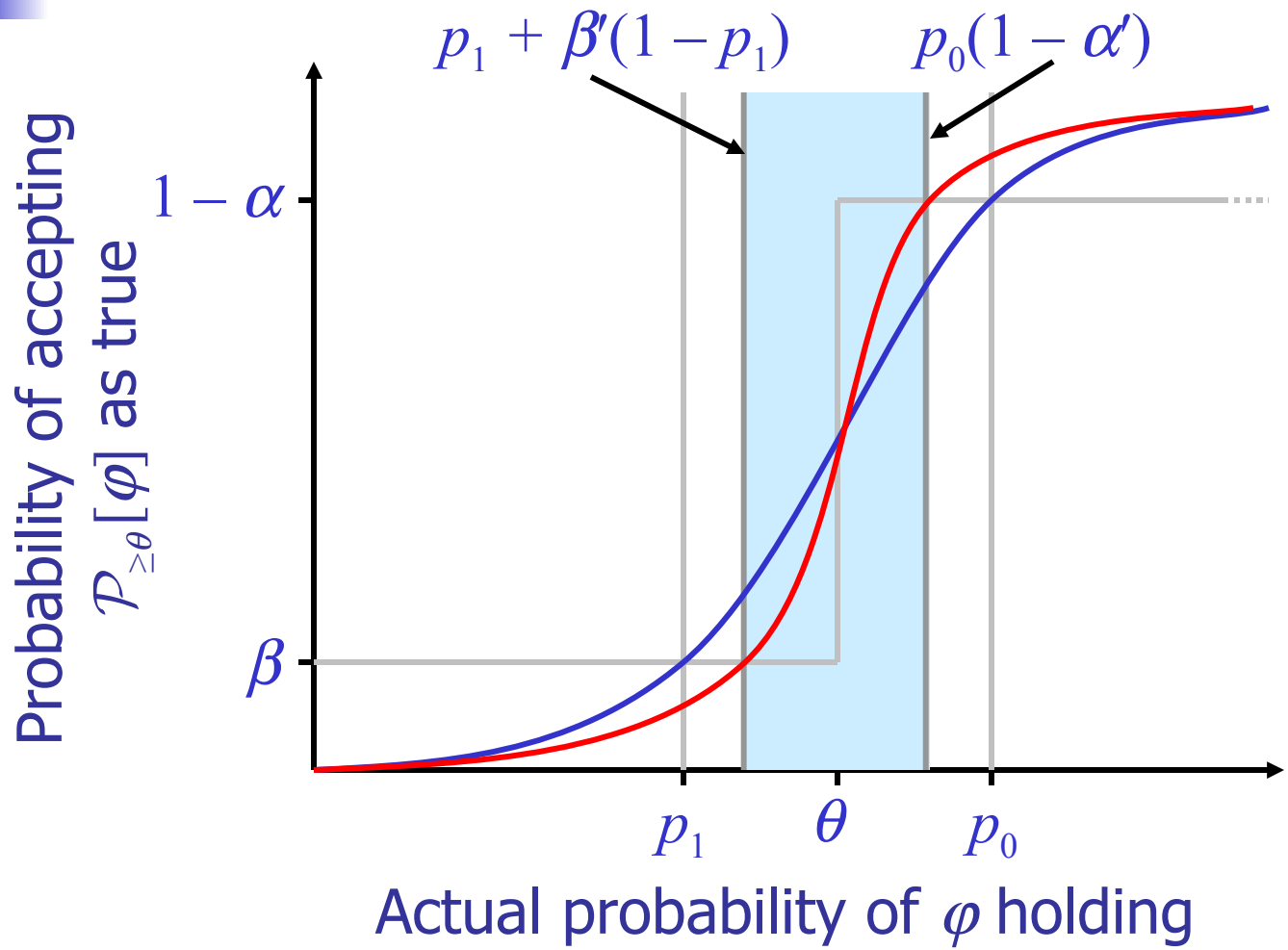


Statistical Verification of Nested Probabilistic Statements

- Cannot verify path formula without some probability of error
  - Probability of false negative: $\leq \alpha'$
  - Probability of false positive: $\leq \beta'$

Observation error

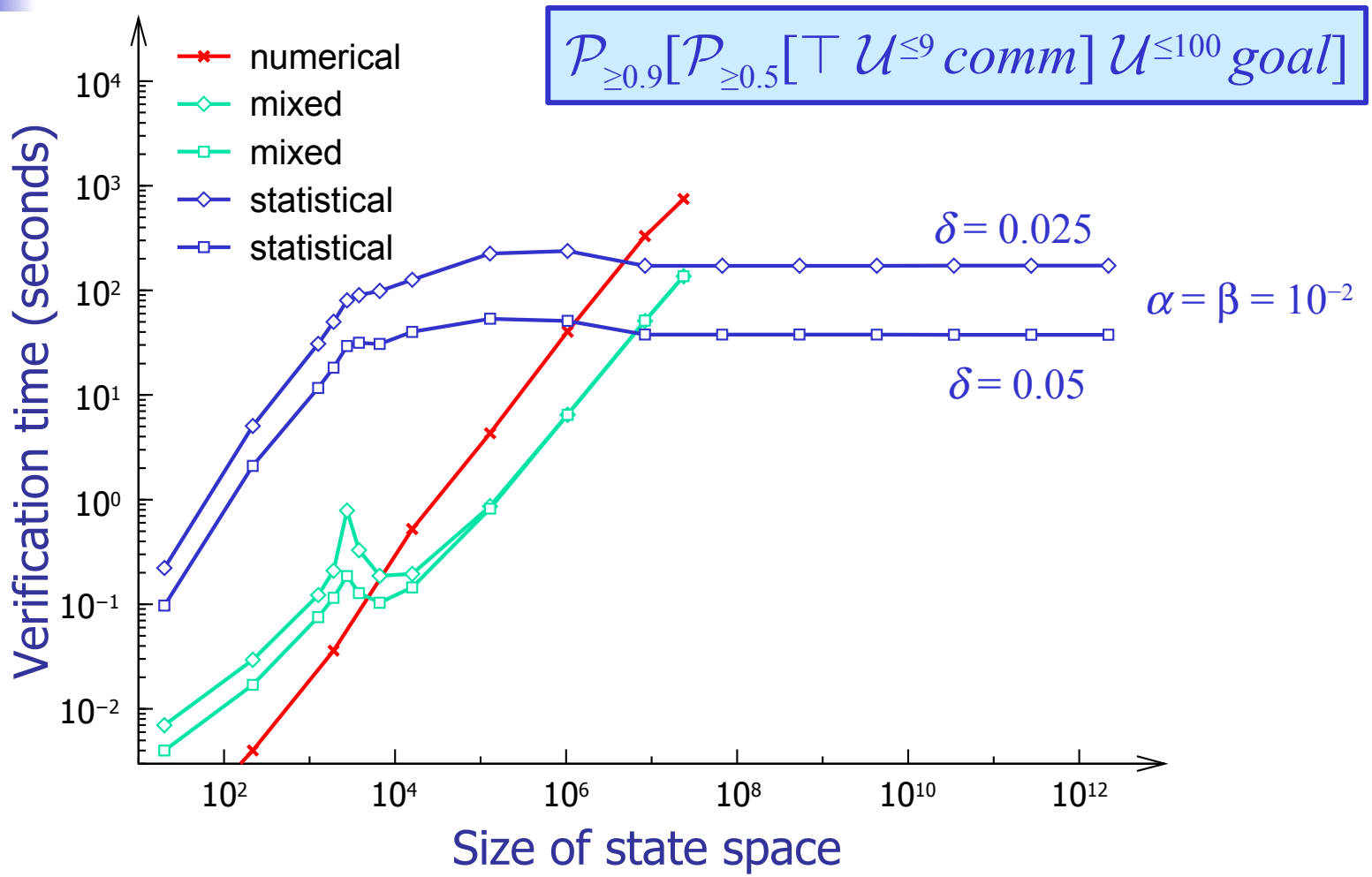
Theorem: Adjusted Indifference Region Handles Observation Error



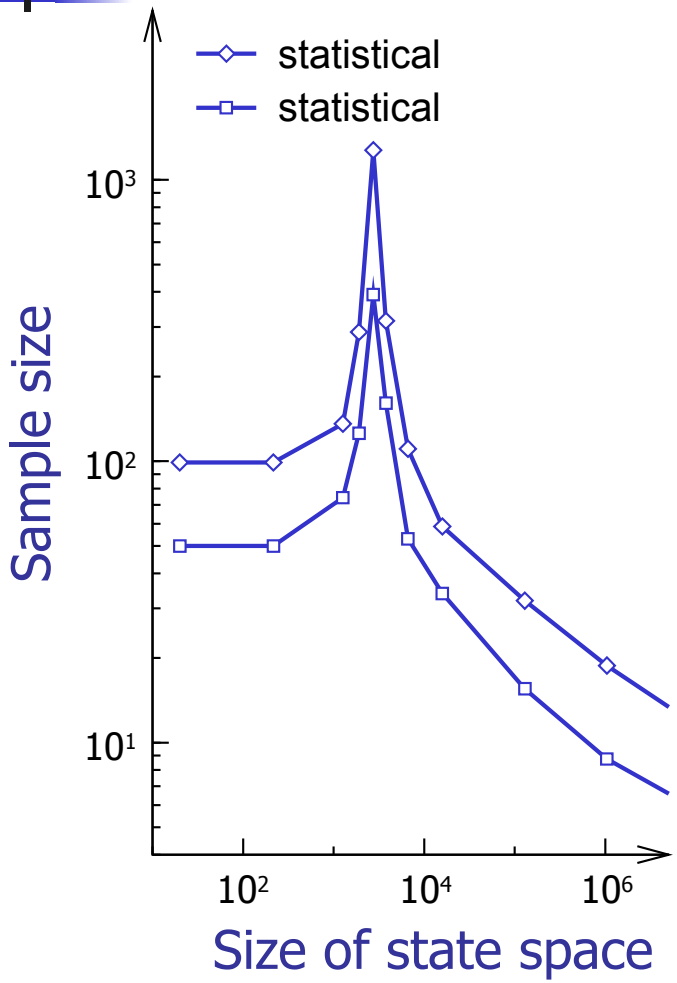
Performance Considerations

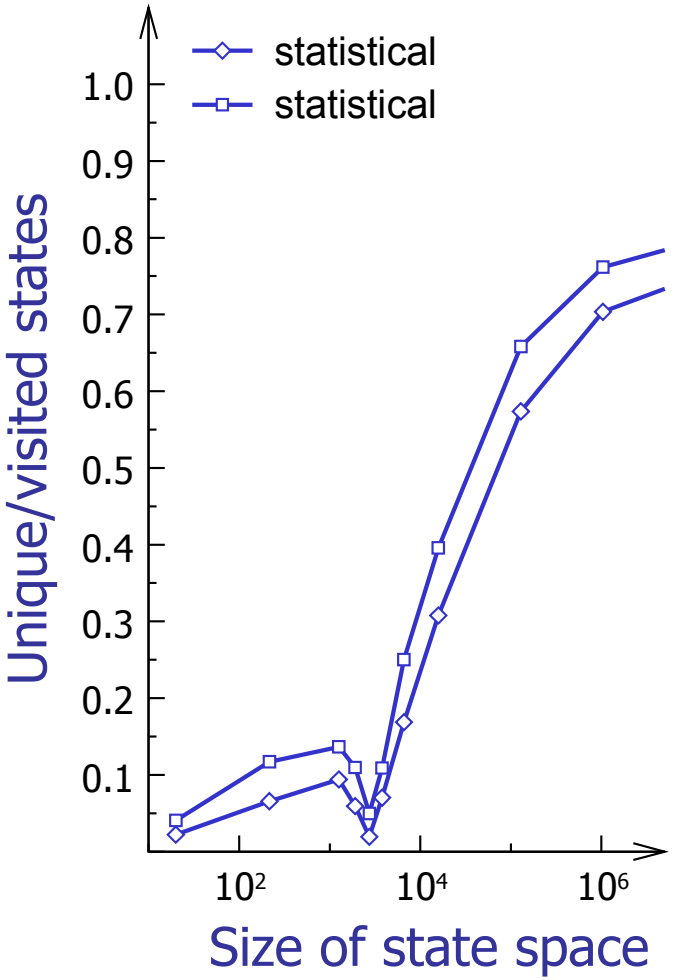
- Verification error is independent of observation error
  - Pick observation error to minimize effort
- The same state may be visited along multiple sample paths
  - Memoize verification results to avoid repeated effort

Robot Grid World (results)



Robot Grid World: Effect of Memoization





Probabilistic Model Checking: Summary

- Acceptance sampling can be used to verify probabilistic properties of systems
- Sequential acceptance sampling adapts to the difficulty of the problem
- Memoization helps in making statistical verification of nested probabilistic operators feasible

Decision Theoretic Planning

- Stochastic model with actions and rewards
  - Generalized semi-Markov decision process
- Objective: Find policy that maximizes expected reward
  - Infinite-horizon discounted reward

A Model of Stochastic Discrete Event Systems

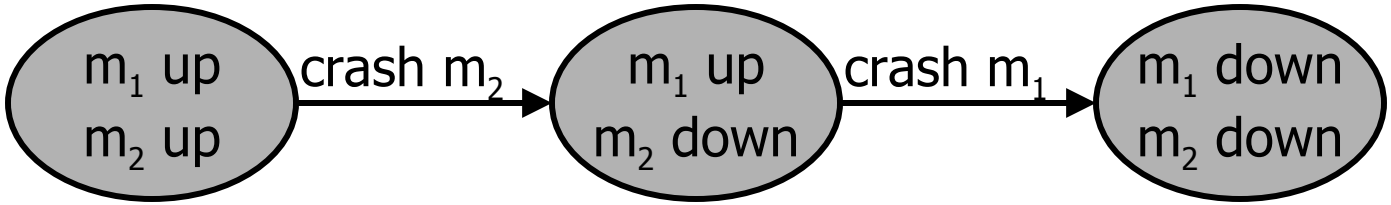
- Generalized semi-Markov process (GSMP) [Matthes 1962]
  - A set of events E
  - A set of states S
- GSMDP
  - Actions $A \subset E$ are controllable events

Events

- With each event e is associated:
  - A condition ϕ_e identifying the set of states in which e is enabled
  - A 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

Events: Example

- Network with two machines
  - Crash time: $Exp(1)$
  - Reboot time: $U(0,1)$



$t = 0$ | $t = 0.6$ | $t = 1.1$

$G_{c1} = Exp(1)$
$G_{c2} = Exp(1)$

$G_{c1} = Exp(1)$
$G_{r2} = U(0,1)$

$G_{r2} = U(0,0.5)$

Asynchronous events $\Rightarrow$ beyond Markov

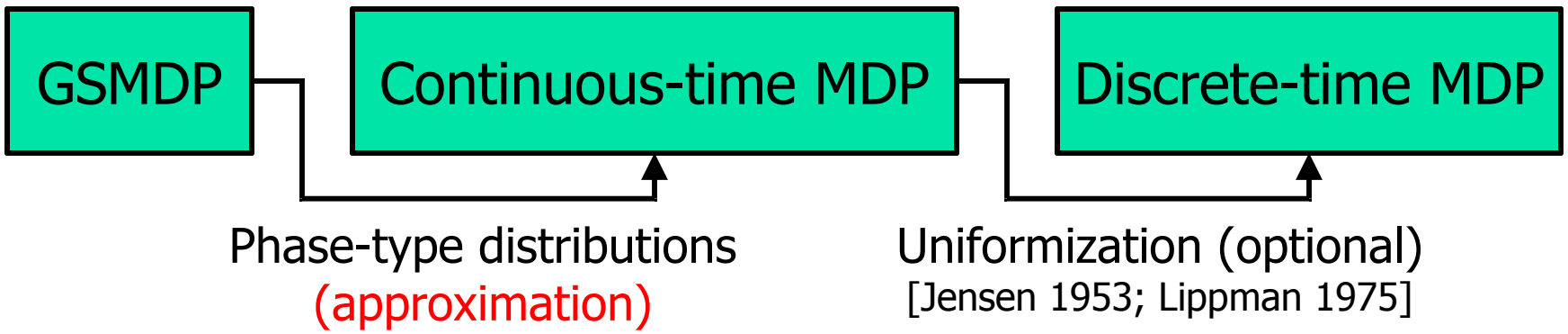
Policies

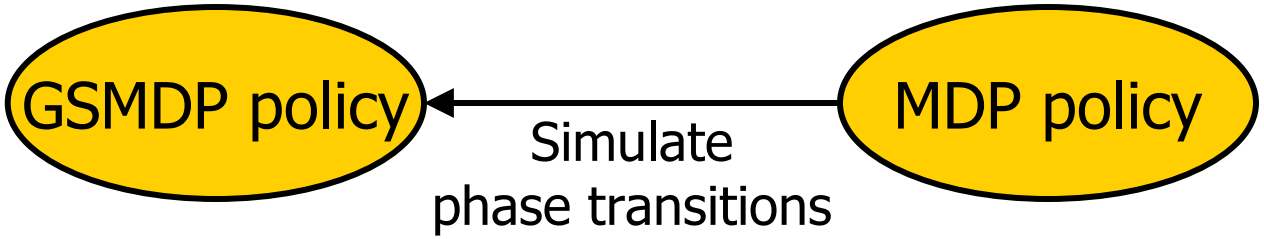
- Actions as controllable events
  - We can choose to disable an action even if its enabling condition is satisfied
- A policy determines the set of actions to keep enabled at any given time during execution

Rewards and Optimality

- Lump sum reward $k_e(s, s')$ associated with transition from s to s' caused by e
- Continuous reward rate $c_{A'}(s)$ associated with A' being enabled in s
- Infinite-horizon discounted reward
  - Unit reward earned at time t counts as $e^{-\alpha t}$
- Optimal choice may depend on entire execution history

GSMDP Solution Method

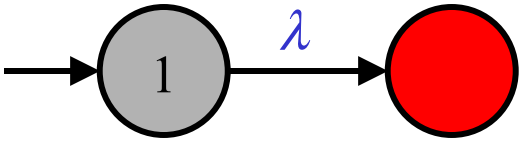




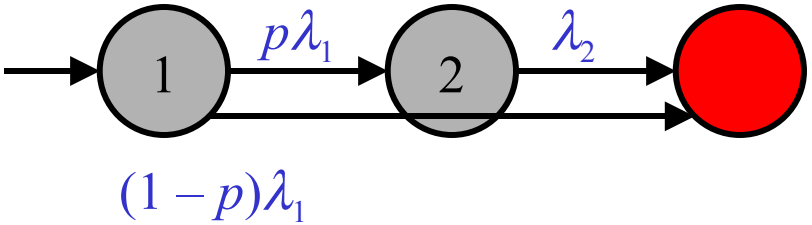
Continuous Phase-Type Distributions [Neuts 1981]

- Time to absorption in a continuous-time Markov chain with n transient states

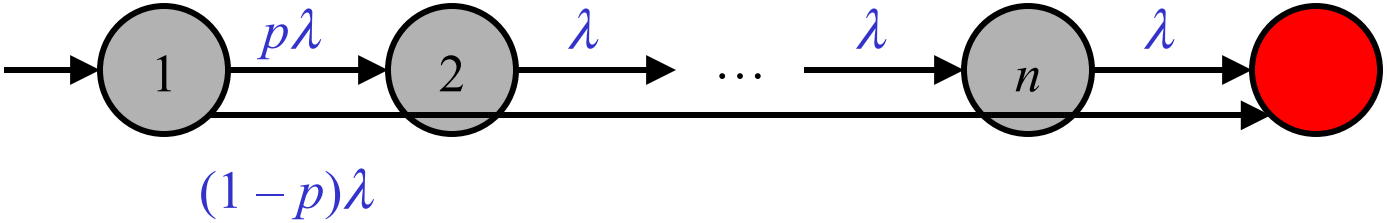
Exponential



Two-phase Coxian



n-phase generalized Erlang



Approximating GSMDP with Continuous-time MDP

- Approximate each distribution G_e with a continuous phase-type distribution
  - Phases become part of state description
  - Phases represent discretization into random-length intervals of the time events have been enabled

Policy Execution

- The policy we obtain is a mapping from modified state space to actions
- To execute a policy we need to simulate phase transitions
- Times when action choice may change:
  - Triggering of actual event or action
  - Simulated phase transition

Method of Moments

- Approximate general distribution G with phase-type distribution PH by matching the first k moments
  - Mean (first moment): μ_1
  - Variance: $\sigma^2 = \mu_2 - \mu_1^2$
  - The ith moment: $\mu_i = \mathrm{E}[X^i]$
- Fast, but does not guaranteed good fit to shape of distribution function

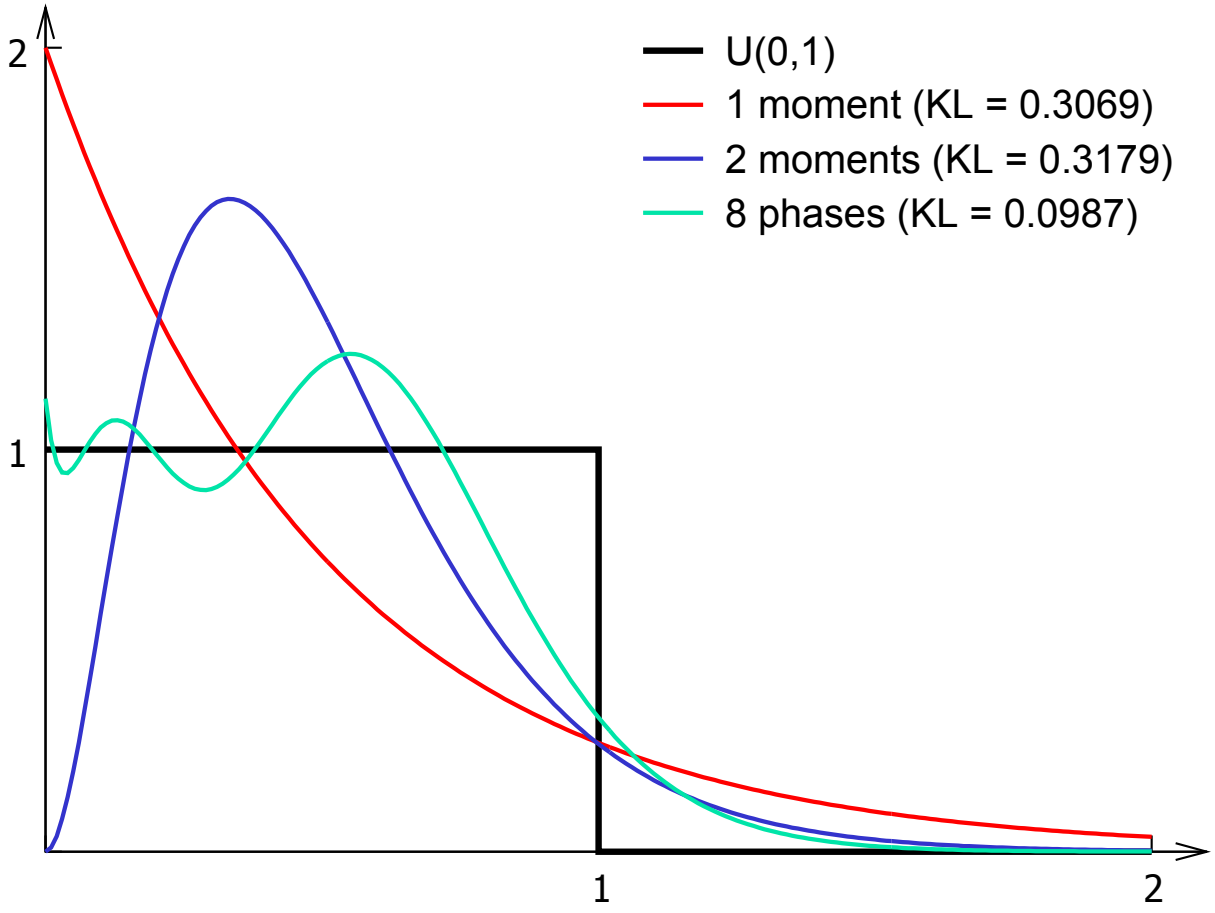
Fitting Distribution Functions [Asmussen et al. 1996]

- Find phase-type distribution with n phases that minimizes KL-divergence of density functions

$$KL(f,g) = \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)} dx$$

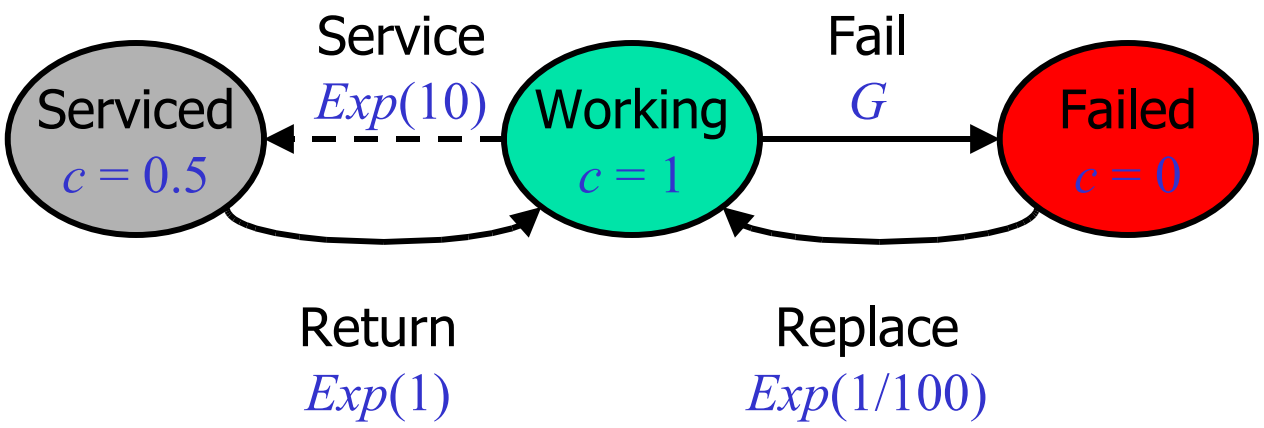
- Slow for large n (EM algorithm)

Phase-type Fitting: Example



The Foreman's Dilemma

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



The Foreman's Dilemma: Optimal Solution

- Find t_0 that maximizes v_0

$$v_0 = \int_0^\infty f_X(t)\left(1-F_Y(t)\right)\left(\left(\frac{1}{\alpha}\left(1-e^{-\alpha t}\right)+e^{-\alpha t}v_1\right)\right)+f_Y(t)\left(1-F_X(t)\right)\left(\frac{1}{\alpha}\left(1-e^{-\alpha t}\right)+e^{-\alpha t}v_2\right)dt$$

$$v_1 = \frac{1}{1+100\alpha}v_0 \qquad v_2 = \frac{1}{1+\alpha}\left(\frac{1}{2}+v_0\right)$$

$$f_X(t) = \begin{cases} 0 & t < t_0 \\ 10e^{-10(t-t_0)} & t \geq t_0 \end{cases}$$

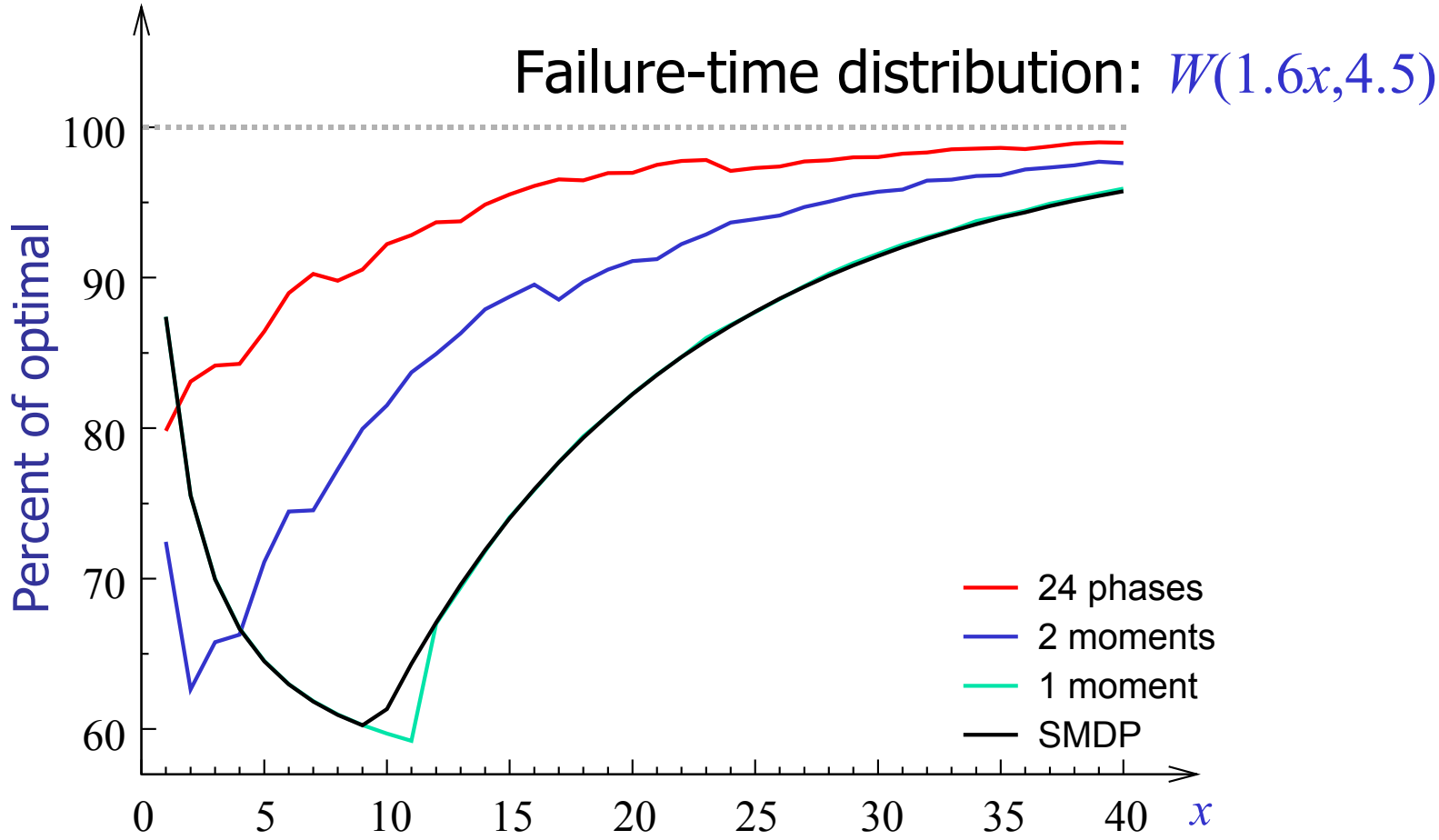
$$F_X(t) = \int_0^t f_X(x)dx$$

Y is the time to failure in "Working" state

The Foreman's Dilemma: SMDP Solution

- Same formulae, but restricted choice:
  - Action is immediately enabled ($t_0 = 0$)
  - Action is never enabled ($t_0 = \infty$)

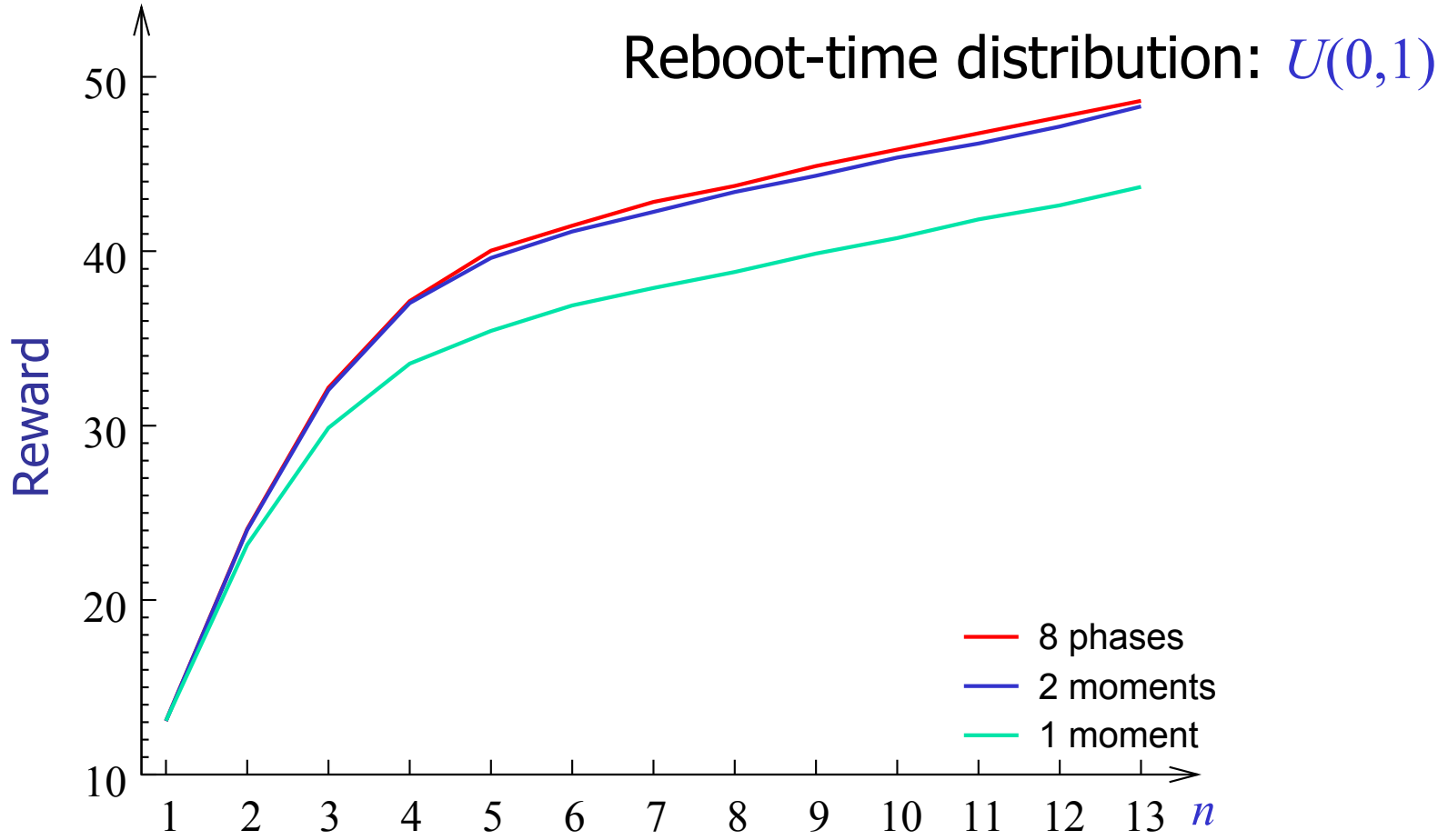
The Foreman's Dilemma: Policy Performance



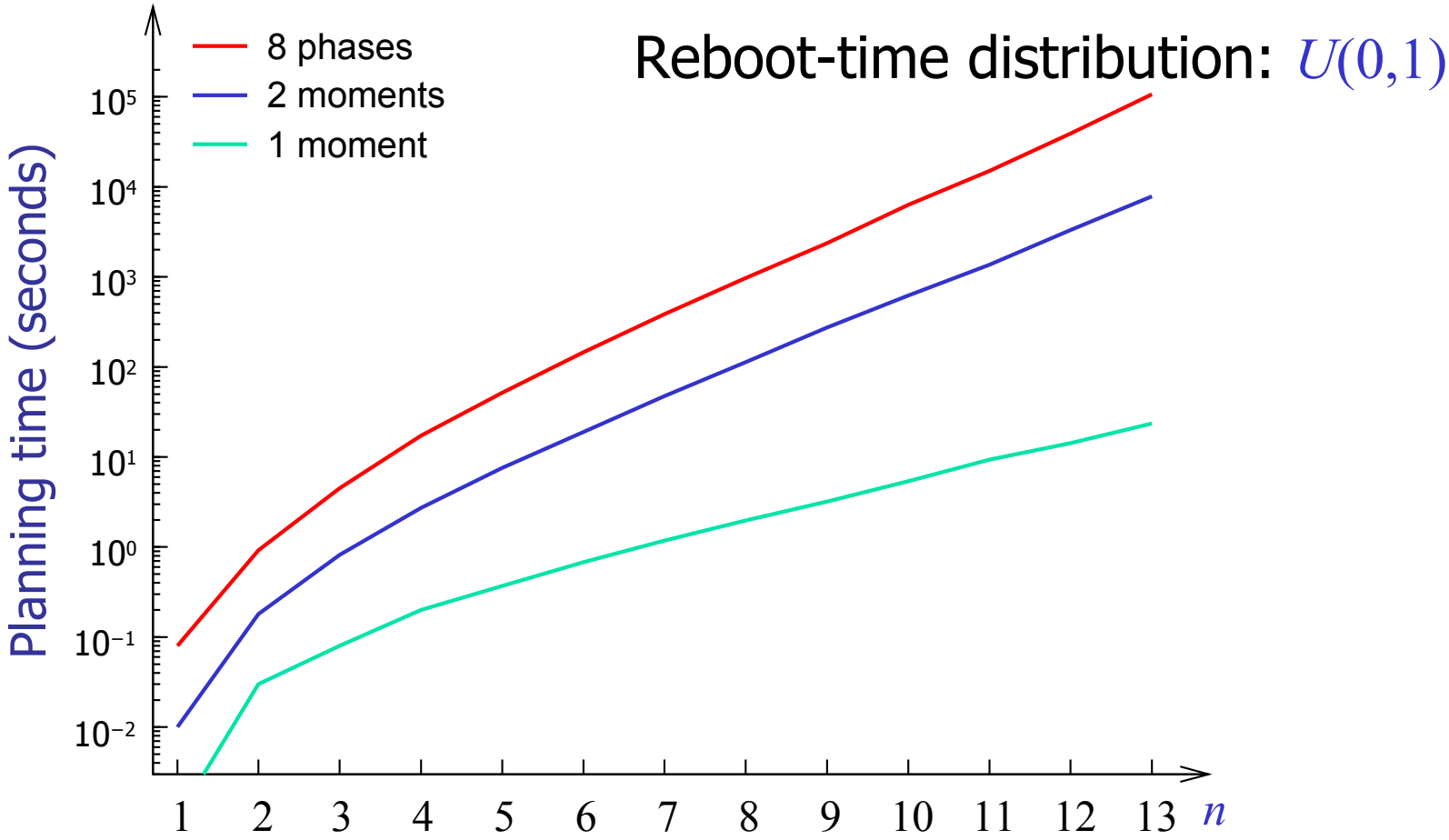
System Administration

- 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



System Administration: Planning Time



Decision Theoretic Planning: Summary

- Phase-type distributions can be used to approximate a GSMDP with an MDP
  - Allows us to approximately solve GSMDPs and SMDPs using existing MDP techniques
- Phase does matter
  - Adding phases often results in higher value
  - Phases permit us to delay enabling of actions or keep actions enabled

Conclusion

- Statistical methods are practical for probabilistic model checking
- Sample path analysis can help in explaining undesirable system behavior
- Phase-type distributions make approximate planning with asynchronous events feasible

Future Work: Verification

- Steady-state properties
  - Use batch means analysis or regenerative simulation with acceptance sampling
- Other acceptance sampling tests
  - Bayesian approach (minimize cost)
- Faster simulation
  - Exploit symbolic data structures
- Applications!

Future Work: Planning

- Goal directed planning
  - Use sample path analysis for mixed initiative planning
- Decision theoretic planning
  - Optimal GSMDP planning
  - Value function approximation
- Applications!

Tools

- Ymer
  - Statistical probabilistic model checking
- Tempastic-DTP
  - Decision theoretic planning with asynchronous events

http://www.cs.cmu.edu/~lorens/

References

Alur, R., C Courcoubetis, and D. L. Dill. 1991. Model-checking for probabilistic real-time systems. In *Proc. ICALP'91*.

Asmussen, S., O. Nerman, and M. Olsson. 1996. Fitting phase-type distributions via the EM algorithm. *Scand. J. Statist.* 23: 419–441.

Atkins, E. M., E. H. Durfee, and K. G. Shin. 1996. Plan development using local probabilistic models. In *Proc. UAI'96*.

Baier, C., B. R. Haverkort, H. Hermanns, and J.-P. Katoen. 2003. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Softw. Eng.* 29: 524–541.

Bellman, R. 1957. *Dynamic Programming*. Princeton University Press.

Bellman, R., R. Kalaba, and B. Kotkin. 1963. Polynomial approximation—a new computational technique in dynamic programming: Allocation processes. *Math. of Comp.* 17: 155–161.

Cantaluppi, L. 1984. Optimality of piecewise-constant policies in semi-Markov decision chains. *SIAM J. Control Optim.* 22: 723–739.

Chitgopekar, S. S. 1969. Continuous time Markovian sequential control processes. *SIAM J. Control* 7: 367–389.

Gordon, G. J. 1995. Stable function approximation in dynamic programming. In *Proc. ICML'95*.

Guestrin, C., D. Koller, and R. Parr. 2002. Multiagent planning with factored MDPs. In *Proc. NIPS'01*.

Guestrin, C., D. Koller, R. Parr, and S. Venkataraman. 2003. Efficient solution algorithms for factored MDPs. *J. Artificial Intelligence Res.* 19: 399–468.

Hansson, H. and B. Jonsson. 1994. A logic for reasoning about time and reliability. *Formal Aspects Comput.* 6: 512–535.

References

Howard, R. A. 1960. *Dynamic Programming and Markov Processes*. John Wiley & Sons.

Howard, R. A. 1963. Semi-Markov decision processes. *Bull. Institut Int. Statistique* 40: 625–652.

Infante López, G. G., H. Hermanns, and J.-P. Katoen. 2001. Beyond memoryless distributions: Model checking semi-Markov chains. In *Proc. PAPM-PROBMIV'01*.

Jensen, A. 1953. Markoff chains as an aid in the study of Markoff processes. *Scand. Aktuar. J.* 36: 87–91.

Kwiatkowska, M., G. Norman, R. Segala, and J. Sproston. 2000. Verifying quantitative properties of continuous probabilistic timed automata. In *Proc. CONCUR'00*.

Lippman, S. A. 1975. Applying a new device in the optimization of exponential queuing systems. *Oper. Res.* 23: 687–710.

Matthes, K. 1962. Zur Theorie der Bedienungsprozesse. In *Trans. Third Prague Conference on Information Theory, Statistical Decision Functions, Random Processes.*

Mausam and Daniel S. Weld. 2004. Solving concurrent Markov decision processes. In *Proc. AAAI'04*.

Musliner, D. J., E. H. Durfee, and K. G. Shin. 1995. World modeling for the dynamic construction of real-time control plans. *Artificial Intelligence* 74: 83–127.

Neuts, M. F. 1975. Probability distributions of phase type. In *Liber Amicorum Professor emeritus dr. H. Florin*. Katholieke Universiteit Leuven.

Stone, L. D. 1973. Necessary and sufficient conditions for optimal control of semi-Markov jump processes. *SIAM J. Control* 11: 187–201.

Wald, A. 1945. Sequential tests of statistical hypotheses. *Ann. Math. Statist.* 16: 117–186.

Phase-type Fitting: Weibull