# A Multi-Trajectory Approach to Rare-Event Simulation

RESIM 2021, Paris

Thomas Hotz[1] and Armin Zimmermann[2]
[1]Institute for Mathematics, Probability Theory and Mathematical Statistics
[2]Computer Science, Systems and Software Engineering
Technische Universität Ilmenau, Germany

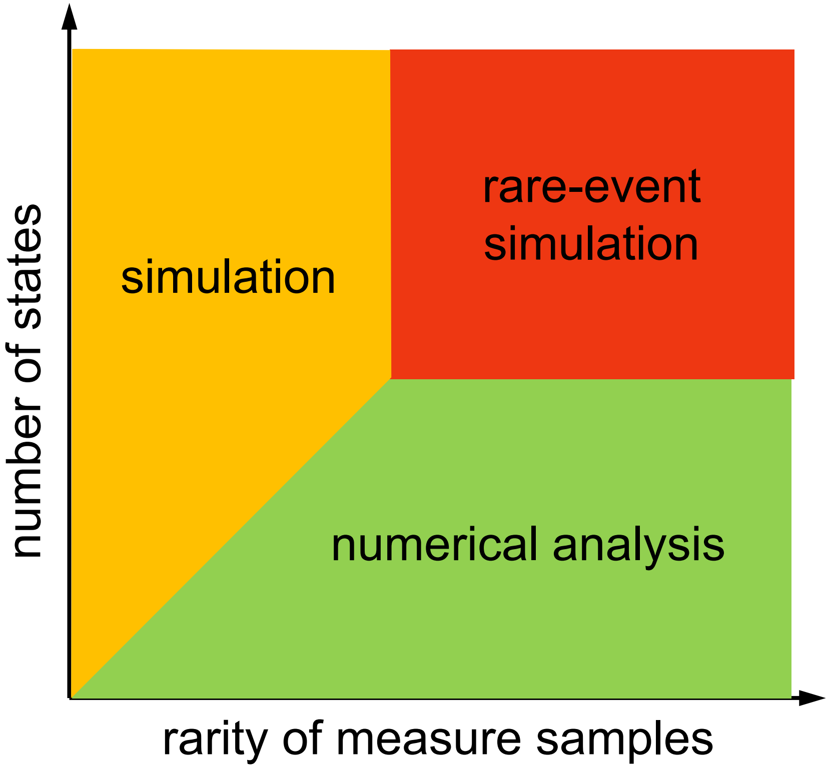
TECHNISCHE UNIVERSITÄT ILMENAU

# Contents

- Performability evaluation methods
- A hybrid multi-trajectory simulation algorithm
- Rare-event application Example
- Conclusion

# Performability Evaluation

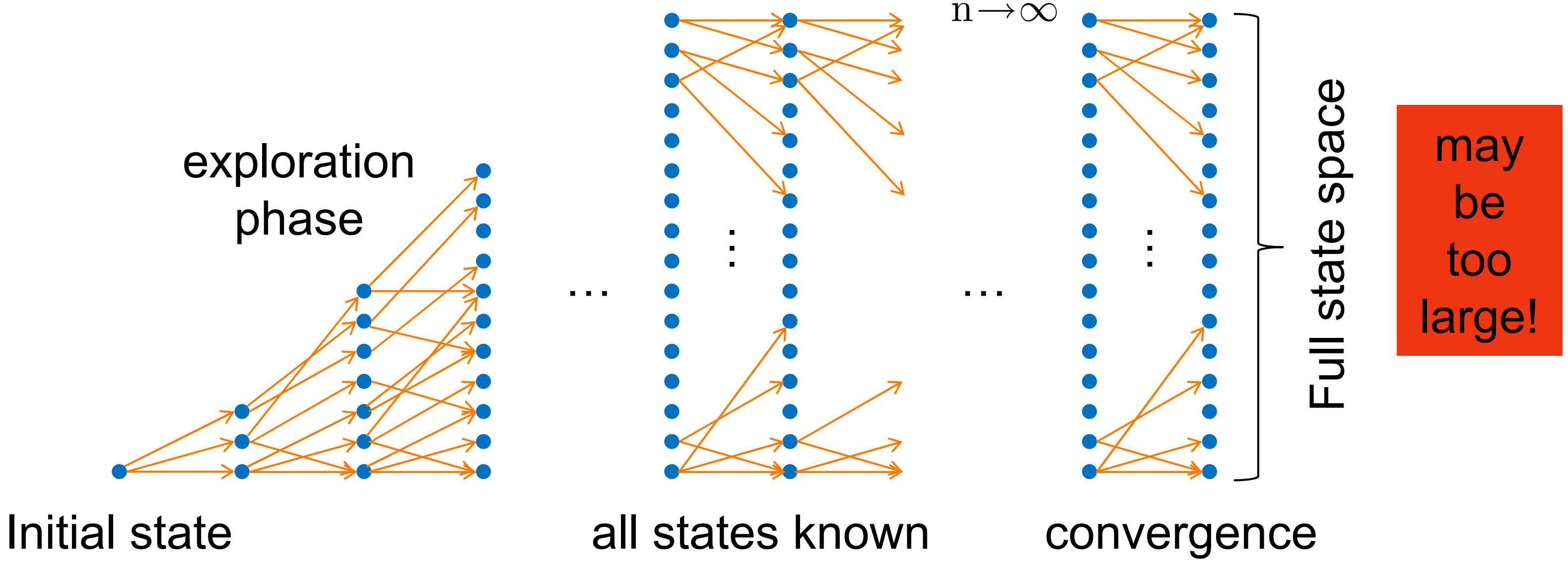
- **Applicability and Efficiency of Algorithms**
  - Two dimensions of (Markovian) problems



# Evaluation Methods

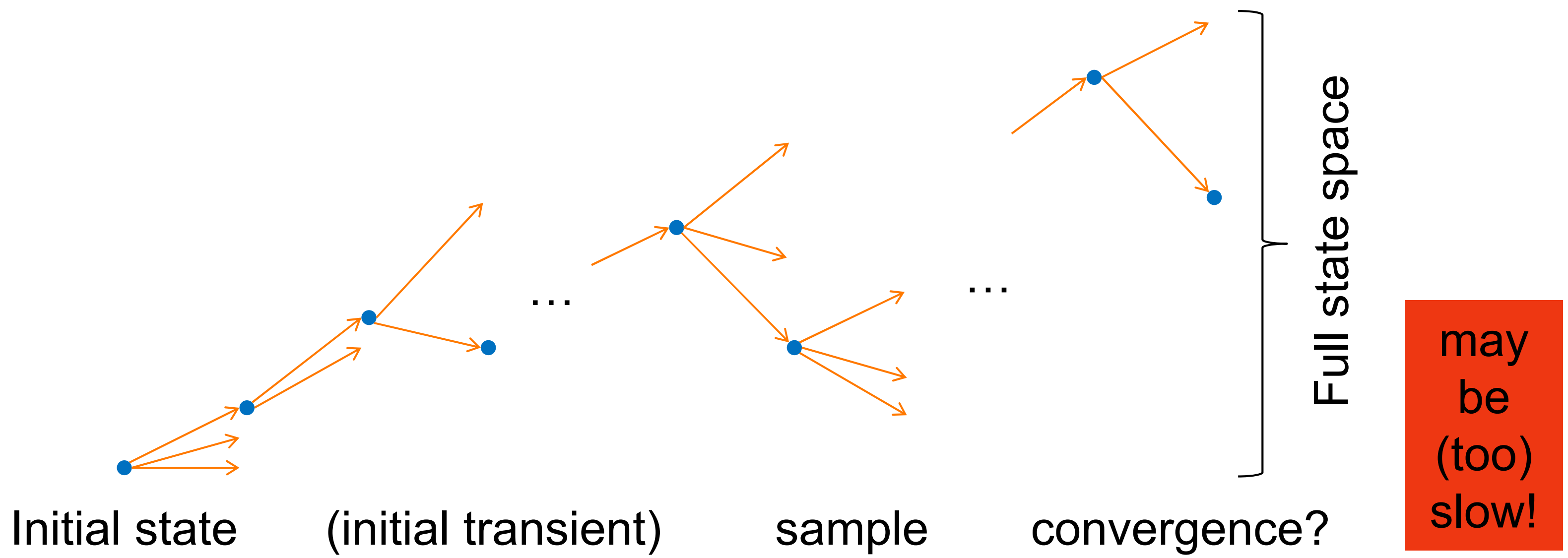
- **Method 1: Numerical Analysis**
  - Based on full state space
  - Derive and solve equations covering stochastic process $X_n$
  - Example: DTMC $p_{ij} = P\left(X_n = j \middle| X_{n-1} = i\right)$

$p(0) \qquad p(i+1) = p(i)\,P \qquad \lim_{n\to\infty} p(n) = \pi$



# Evaluation Methods

- **Method 2: Simulation**
  - Visit states sequentially
  - Choose next possible state randomly
  - Collect samples on the fly
  - Derive stochastic estimator (mean, variance)



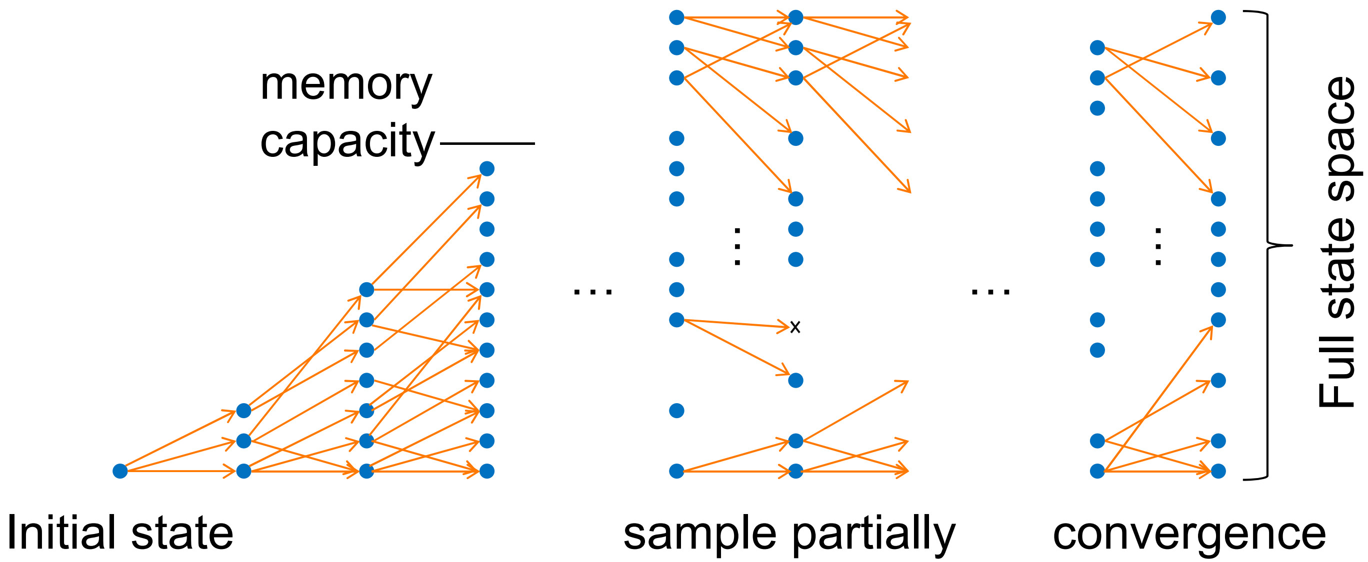
# Hybrid Multi-Trajectory Simulation

- **An Integrated Algorithm**
  - Keep advantages of simulation *and* numerical analysis
  - Work like numerical analysis, as long as memory suffices
    - **"split" particle step** following all possible state changes
    - distribute state probability over destination states
  - Select state transition like simulation if not
    - **probabilistic step** of trajectory
    - merge trajectories leading to stored state (add $p_i$ s)
    - … other heuristics are possible (adaptive number of particles..)

  - Sum of state (particle) probabilities will always be 1
    - $p_i$ equals probability that a trajectory would have arrived at state i under the previous decisions

# Hybrid Multi-Trajectory Simulation

- **General Idea**
  - Keep as many states as possible
  - Mix simulation and numerical analysis steps
  - Treat continuous-time models via DTMC embedding



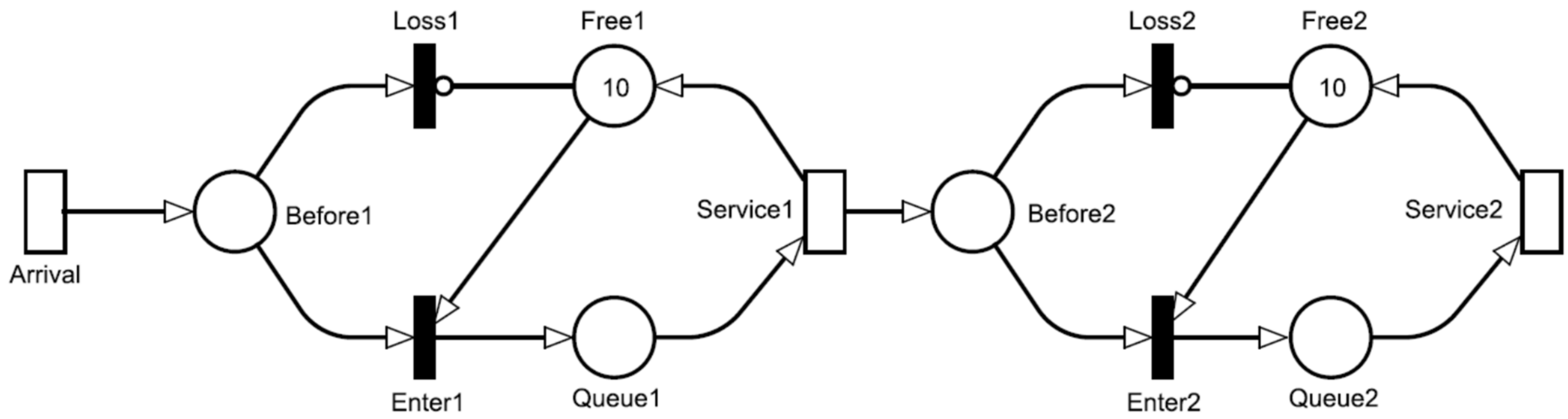
# Hybrid Multi-Trajectory Simulation

- **Proofs in Previous Papers**
  - Unbiasedness
  - (Almost sure) convergence, at least as fast as simulation
  - Based on a **unified vector-matrix framework** treating numerical analysis, simulation, and multi-trajectory in the same way

- **Contains Standard Methods as Special Cases**
  - Numerical analysis: $\left|\text{Particles}\right| \geq \left|\text{ReachableStates}\right|$
  - Simulation: $\left|\text{Particles}\right| = 1$

# Application Example

- **Tandem Queue Model**
  - Stochastic Petri net, customer losses (no blocking)
  - Service rate 2 > service rate 1 > arrival rate




  - Probability of #customers > $n$ in $2^{nd}$ queue: rare event

# Application Example

- **Tandem Queue Model**
  - Results
    averaged over several runs, with 95% conf. interval / 5% relative error

| Arrival | Numeric analysis | | | Simulation | | | Adaptive Multi-Traj. | | |
|---|---|---|---|---|---|---|---|---|---|
| Delay | Result | Time | Mem. | Error | Time | Mem. | Error | Time | Mem. |
| 1.0 | 1.149E-3 | 0.1 | 128 | 1.9% | 1.2 | 28 | 0.6% | 0.9 | 241 |
| 1.5 | 1.143E-4 | 0.1 | 128 | 1.8% | 6.9 | 30 | 0.7% | 0.8 | 240 |
| 2.0 | 1.378E-5 | 0.1 | 128 | 0.2% | 35.9 | 76 | 0.8% | 0.8 | 241 |
| 2.5 | 2.442E-6 | 0.1 | 128 | 0.3% | 203.4 | 66 | 0.6% | 0.8 | 240 |
| 3.0 | 5.804E-7 | 0.1 | 128 | 0.9% | 767.5 | 66 | 0.9% | 0.8 | 240 |
| 3.5 | 1.708E-7 | 0.1 | 128 | 1.7% | 2 239.4 | 80 | 0.8% | 0.8 | 240 |
| 4.0 | 5.902E-8 | 0.1 | 128 | 4.4% | 4 476.3 | 56 | 0.9% | 0.8 | 241 |
| 4.5 | 2.308E-8 | 0.1 | 128 | | | | 0.5% | 0.7 | 244 |
| 5.0 | 9.953E-9 | 0.1 | 128 | | | | 0.5% | 0.7 | 244 |

  - Accuracy, computation time and memory consumption

# Conclusion

- **Multi-Trajectory Simulation**
  - A hybrid performability evaluation algorithm integrating elements of numerical analysis and simulation
    - Mathematical framework proposed
    - Efficient compared to standard methods, useful for rare-event problems
    - Extension of splitting methods
  - Prototype tool implementation in TimeNET

  - Future work
    - Heuristics for rare-event problems with large state space
    - Extension to non-Markovian models

# Further Information

- **Tool**
  - http://timenet.tu-ilmenau.de/
- **Background and Details**
  - A. Zimmermann and T. Hotz: *Integrating simulation and numerical analysis in the evaluation of generalized stochastic Petri nets*, ACM TOMACS 29(4) 2019.
- **Application**
  - A. Zimmermann, T. Hotz, V. Hädicke, and M. Friebe: *Analysis of safety-critical cloud architectures with multi-trajectory simulation*, accepted for publication at RAMS 2022.

- **Algorithm**

**repeat** (* main simulation loop *)
  **while** $|Particles| > 0$ **do**
    Select any $p \in Particles$ ; $Particles := Particles \setminus \{p\}$
    $w := p.weight$ ; $m := p.marking$
    $\mathcal{T}_{ena}$ := set of all transitions enabled in marking $m$

    (* rate reward and sojourn time *)
    $WeightSum := \sum_{tr \in \mathcal{T}_{ena}} \lambda(tr)$
    $Reward \mathrel{+}= \frac{w}{WeightSum} RateReward^{rvar}(m)$
    $SimTime \mathrel{+}= \frac{w}{WeightSum}$

    (* decision heuristic, here: only split if enough space *)
    **if** $|Particles| + |Particles'| + |\mathcal{T}_{ena}| > N$ **then** (* don't split *)
      Select any $tr \in \mathcal{T}_{ena}$ randomly
      $\mathcal{T}_{ena} := \{tr\}$ ; $WeightSum := \lambda(tr)$

    (* fire transition(s) *)
    **for** $\forall tr \in \mathcal{T}_{ena}$ **do**
      $m' := FireTransition(m, tr)$
      **if** $m' \notin Particles'$ **then** (* add new particle *)
        $Particles' := Particles' \cup \{(m', 0)\}$
      (* merge particle weight *)
      $Particles'(m').weight \mathrel{+}= \frac{w}{WeightSum} \lambda(tr)$
      (* impulse reward for fired transition *)
      $Reward \mathrel{+}= \frac{w}{WeightSum} ImpulseReward^{rvar}(m, tr)$

  (* full particle set finished *)
  $Particles := Particles'$ ; $Particles' := \{\}$
  Collect measure sample with value $\frac{Reward}{SimTime}$