TUHH
Hamburg University of Technology

Rare event analysis using the Limited Relative Error Algorithm for OMNeT++ simulations

Sebastian Lindner, Raphael Elsner, Phuong Nga Tran and Andreas Timm-Giel

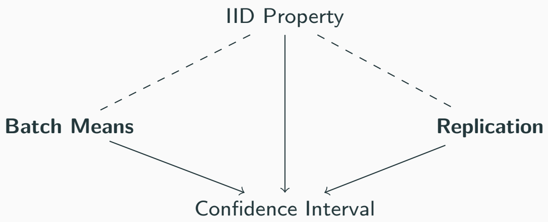
OMNeT++ Summit, 6th and 7th of September, 2018

Institute for Communication Networks

Table of contents

Motivation

Stochastic simulation → *statistical evaluation* → objective statement



Stochastic simulation $\rightarrow$ *statistical evaluation* $\rightarrow$ objective statement

Evaluation methods using Confidence Intervals

1. **Batch Means**
   1 simulation run $\rightarrow (x_1, x_2, \ldots, x_n)$ observations
   $\rightarrow$ split into k batches of b observations ($n = kb$)
   $\rightarrow$ find batch means $Y_i(b)$
   $\rightarrow$ *reduce* sample correlation by forming "quasi-independent, quasi-normally distributed batch-random variables"
   "deficient" according to [1]
   $\rightarrow$ what's the right batch length and simulation time?

Stochastic simulation $\rightarrow$ *statistical evaluation* $\rightarrow$ objective statement

Evaluation methods using Confidence Intervals

2. **Replication method**
   i simulation runs $\rightarrow [(x_{1,1}, x_{2,1}, \dots, x_{n,1}),\ \dots,\ (x_{1,i}, x_{2,i}, \dots, x_{n,i})]$
   $\rightarrow$ i mean values, one per repetition
   $\rightarrow$ repetition of same scenario eliminates correlation
   $\rightarrow$ have to eliminate warmup period
   $\rightarrow$ runs need to be long enough to be iid

Motivation, cont.

- How do you know *a-priori*
  - how many observations or repetitions are required
  - what the simulation time should be

  for a *statistically sound* analysis?

$\Rightarrow$ Akaroa2, from [2]:
  - runs distributed simulations
  - merges results centrally
  - analyses results online
  - stops processes once results are deemed confident enough

- Confidence intervals break for very rare and very likely events

Limited Relative Error (LRE) attempts to

(a) approximate an unknown cumulative distribution function (CDF) function $F_X(x)$ as $\widetilde{F}_X(x)$,
(b) make statements about the sample sequence correlations,
(c) determine a *relative error* function,
(d) request more samples until an error bound is met,
(e) requires a single simulation run and monitors sample correlation,
(f) is designed to work well with very rare events.

Confidence interval-based methods evaluate the *mean* of a statistic

$\Rightarrow$ suited to obtain a picture of the *range* of a statistic

fails for very rare / likely events (Normal distribution assumption doesn't hold)

"What is the average packet delay this system achieves?"

LRE evaluates the *distribution* of a statistic

$\Rightarrow$ suited for *reliability analysis*

can specify target resolution and max. accepted error beforehand

"How likely is it that this system experiences VoIP packet delays > 150 ms"

Algorithm Description

Markov chains in LRE

Figure 1: Graphical visualization of $G(x)$ in Equation 1.

1. Obtain observations $(x_1, x_2, \ldots, x_n)$
2. $(x_1, x_2, \ldots, x_n)$ corresponds to $(k+1)$-state Markov chain
3. For this, the complementary cumulative distribution function (CCDF) $G(x)$ can be found as

$$G(x) = G_i = P(X > x) = \sum_{j=i}^{k} P_j \text{ for } i-1 \leq x < i,\ i = 1, 2, \ldots, k \quad \text{with } G_0 = 1 \text{ and } G_{k+1} = 0 \tag{1}$$

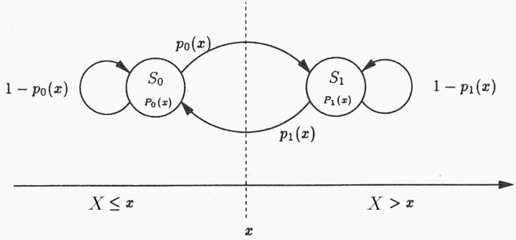
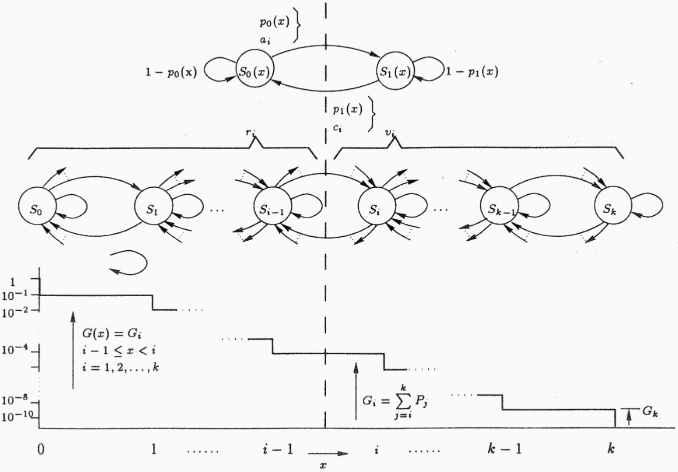


Figure 2: A local x-based 2-state Markov chain obtained from a $(k+1)$-state Markov chain for any position x. From [3].

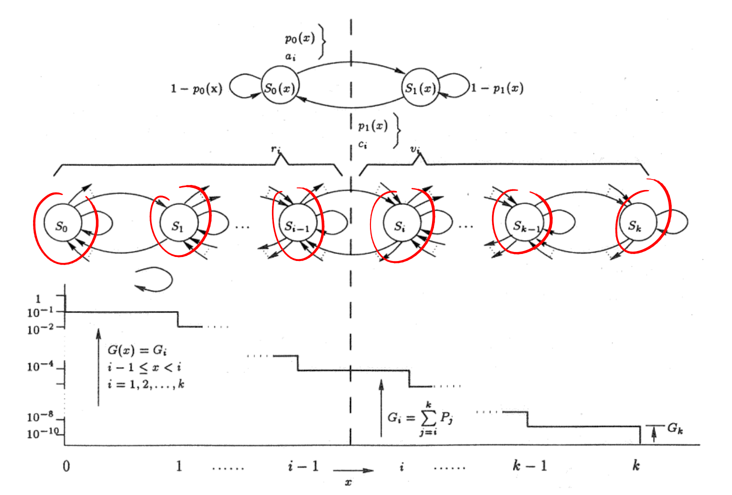
The *local correlation coefficient* can be found as

$$\rho(x) = 1 - (p_0(x) + p_1(x)) \tag{2}$$

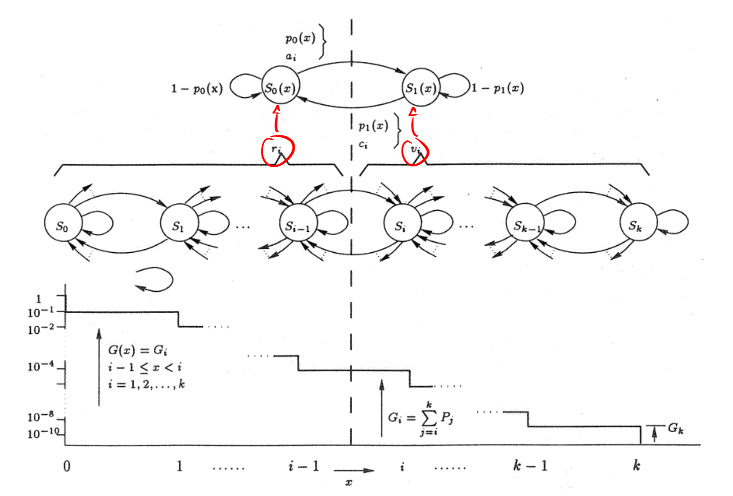
1. Goal: determine the CCDF $G(x)$ where transition probabilities p_{ij} are initially *not* known.



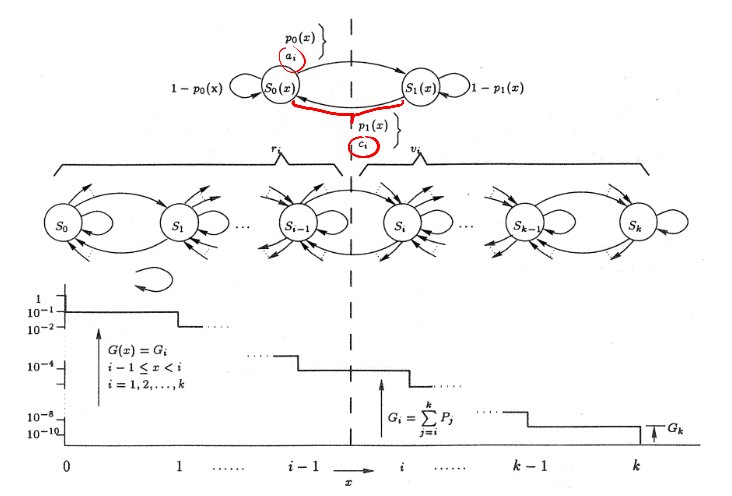
2. Count how many times each state has been entered in counter h_i after n transitions.



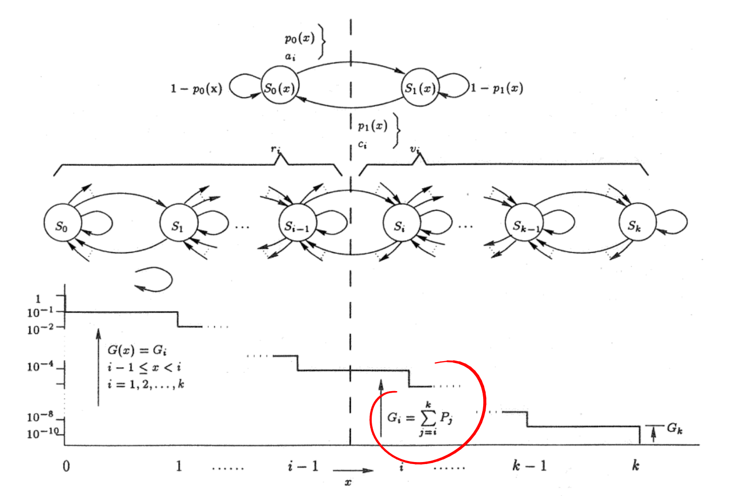
3. Find right state $S_1(x)$ frequency $v_i = \sum_{j=i}^{k} h_j$ for $i = 0, 1, \ldots, k$, $v_0 = n$ left state $S_0(x)$ frequency $r_i = n - v_i$.



4. Count $S_1(x) \to S_0(x)$ transition frequency in c_i (a_i in an analogue way).



5. We can now find $\widetilde{G}(x) = \frac{v_i}{n}$, $\widetilde{\rho}(x)$ and the *relative error* $d_i = \sigma_{G(x)}/\widetilde{G}(x)$.



The relative error d_i

$$d_i = \frac{\sigma_{G(x)}}{\widetilde{G}(x)} \mathrel{\widehat{=}} \text{confidence of results}$$

- $\sigma_{G(x)}$ is the (normally distributed [4] [5]) standard deviation of $S_1(x)$ in the 2-state Markov chain
  and a function of the *correlation factor* $\widetilde{cf}(x) = \frac{1+\widetilde{\rho}(x)}{1-\widetilde{\rho}(x)}$
- $d_i = \frac{\text{absolute error}}{P(\text{in state} \geq i)} \Rightarrow$ relative error
- many transitions from a state to itself $\Rightarrow$ large $\widetilde{\rho}(x) \Rightarrow$ large d_i

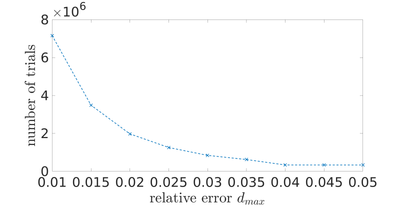
$\Rightarrow$ algorithm demands more observations to ensure accurate modeling of transitions *between* states

Simulation ends when $d_i \leq d_{max} \; \forall i = 0, \ldots, k$.

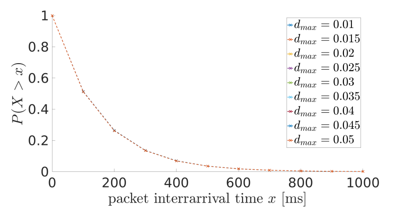
Usage

- The LRE algorithm as a standalone version is available as open-source at [6] (based on openWNS simulator [7]).
- The OMNeT++ integration is available as open-source at [8].
- A novel `LRE` entity can be added to network models.
  - It is easily configured in the `.ini file`.

```
**.lre.xmin = 0.0
**.lre.xmax = 1.0
**.lre.bin_size = 0.1
**.lre.max_error = ${e=0.01..0.05 step 0.005} # usually just 1 value
**.lre.evaluation_interval = 1000
**.lre.output_file = "lre_output_evaluation_e${e}.txt"
```



(a) Number of observations LRE requested for different d_{max}.



(b) CCDF LRE computed for different d_{max}.

- LRE is an alternative method to determine the confidence of simulation results.
- LRE determines when a simulation should end to obtain confident results *in the desired range*.
- The intended resolution of the statistic must be input a-priori.

⇒ It is suited for reliability analysis, where known performance bounds can be tested.

- It is convenient to use: easy configuration, single run, no post-processing.
- A new algorithm description has been given.
- The algorithm is made available as open-source both standalone and as an OMNeT++ integration.
- The combination with the RESTART method ([9] [10]) could reduce simulation time for very rare events could prove very useful to researchers.

Thank you very much for your attention! :)

[1] F. Schreiber and C. Görg, "Stochastic Simulation: A Simplified LRE-Algorithm for Discrete Random Sequences," *AEÜ - International Journal of Electronics and Communications*, 1996.

[2] G. C. Ewing and K. Pawlikowski, "Akaroa2: Exploiting Network Computing by Distributing Stochastic Simulation." SCSI Press, 1998.

[3] C. Görg, "Verkehrstheoretische Modelle und Stochastische Simulationstechniken zur Leistungsanalyse von Kommunikationsnetzen," Habilitation, RWTH Aachen, Germany, 1997.

[4] N. T. Müller, "An Analysis of the LRE-Algorithm using Sojourn Times," *ESM*, 2000.

[5] F. Schreiber, “Reliable Evaluation of Simulation Output Data: a simplified Formula Basis for the LRE-Algorithm,” in *MMB*, 1999.

[6] “LRE Implementation.” [Online]. Available: https://doi.org/10.5281/zenodo.1312970

[7] “openWNS.” [Online]. Available: https://launchpad.net/openwns

[8] “LRE OMNeT++ Integration.” [Online]. Available: https://doi.org/10.5281/zenodo.1313054

[9] G. Carmelita and S. Friedrich, “The RESTART/LRE Method for Rare Event Simulation,” in *Proceedings of the 1996 Winter Simulation Conference*, J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, Eds., 1996.

[10] V.-A. Manuel and V.-A. José, “RESTART: A Method For Accelerating Rare Event Simulations,” *ITC*, 1991.