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

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

Institute of Communication Networks (ComNets)
Hamburg University of Technology (TUHH)
Hamburg, Germany
{sebastian.lindner, raphael.elsner, phuong.tran, timm-giel}@tuhh.de

Abstract

The Limited Relative Error algorithm is an alternative statistical method for data evaluation. Through online result analysis it continuously requests more samples until it deems the evaluation confident enough. With this it allows researchers to hand over the control of simulation time to the algorithm, and through a-priori configuration the target result resolution is set so that arbitrarily rare events can be investigated. We provide a new description of the method as well as a stand-alone implementation and an integration of the algorithm into the OMNeT++ simulator.

1 Introduction

A crucial part of stochastic simulation is the statistical evaluation of the result data. To make an objective statement about the accuracy of one's results, the quantity of data or equivalently the simulation time must be taken into account.

The popular *Batch Means* evaluation method is, according to [1], deficient as it attempts to eliminate correlation by forming "quasi-independent, quasi-normally distributed batch-random variables". The replication method eliminates correlation through the repetition of the same scenario with varying random number generator seeds. This suffers from having to eliminate the warmup period of each repetition. Both methods have the disadvantage of having to estimate the required simulation time a-priori. Akaroa2, presented in [2], aims to solve this problem by running distributed, statistically independent simulations on isolated hosts or processes, which report their progress to a central analyser, which in turn requests the hosts to stop once it deems the results confident enough.

The lesser-known Limited Relative Error (LRE) algorithm will be reviewed in this paper, which estimates an unknown cumulative distribution function (CDF) function corresponding to the observations made during simulation. The algorithm does an online analysis of the simulation results *during* simulation, and so *controls* the simulation length until the *relative error* d has been decreased past the target error $d \leq d_{max}$ by requesting more observations. d is a function of the observation correlations and so, in contrast to Batch Means, LRE monitors the observed correlation and adapts the simulation time according to it.

In contrast to Akaroa2, the distribution of a statistic in question is evaluated, as opposed to the mean. Therefore LRE tackles a different problem and is especially suitable for applications in reliability analysis, where the user knows a-priori that a system *should* not exceed some performance boundaries; say Voice over IP (VoIP) applications should not experience packet delays exceeding 150 ms, and the researcher aims to find out how likely it is to violate this boundary. On the other hand, analysis methods for mean values based on confidence intervals are better suited to establish a picture of the range of some statistic.

In certain applications, very rare events want to be investigated. Very long simulation runs are required to obtain a sufficient number of observations, and time constraints make this a difficult problem, especially if emulators instead of simulators are used. Estimating the required simulation time to obtain a sufficient number of observations is especially difficult in this case.

The paper is organized as follows. In Section 2 a brief history of the LRE algorithm is given. Section 3 attempts to describe the algorithm in detail. Section 4 gives a quick example of the application of the algorithm on work conducted using the OMNeT++ simulator, and Section 5 concludes the document.

2 Related Work

The LRE algorithm evolved over the span of more than a decade. In 1984 Schreiber published the first version in [3], where a sequence of observations was assumed to be independent. In 1988, Schreiber published an extension in [4], where the correlation of the observations was considered. In 1996, Schreiber and Görg published the third iteration of the algorithm in [1], which is further discussed in Görg's habilitation in [5]. In [6] it is evaluated for reliability, while [7] gives an analysis of the algorithm from the perspective of sojourn times, where a disadvantage of inaccurate confidence is highlighted in certain cases, as well as doubt raised about the modelling of samples through a *memoryless* Markov chain, which may be unrealistic. However the advantage of not having to estimate the simulation time may outweigh this disadvantage.

LRE-III is a simplified version that requires the observations to be *discrete*, while the earlier versions were able to also cope with continuous and mixed continuous-discrete observations.

3 Limited Relative Error algorithm

This section covers the functionality of the LRE algorithm as taken from the various publications by the authors in [1], [3], [4] and [5].

3.1 Assumptions and Goals

Given a chronological sequence of observations $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ that stem from a random process X, the following assumptions are typically made for the field of communication networks, from [5]: (a) the samples are *correlated*, (b) the random process X is *stationary*; that is, a time-independent CDF $F_X(x)$ is associated with the sample sequence and (c) the *type* of the random process is unknown; that is, all of $F_X(x)$ as well as correlations are not known a-priori.

The goal of the LRE algorithm is therefore to (a) find $\widetilde{F}_X(x)$ as an approximation of the unknown $F_X(x)$, (b) make statements about the correlation of the sample sequence, (c) determine an error function which can be reduced by adding more samples into the sequence until an error bound is met.

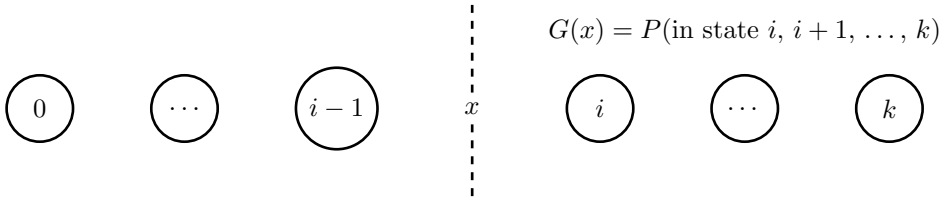


Figure 1: Graphical visualization of $G(x)$ in Equation 1. i is the first state that corresponds to a sample whose value is larger than x.

3.2 Complementary cumulative distribution function

We can order our observations into $(\alpha'_1, \alpha'_2, \ldots, \alpha'_n)$ with $\alpha'_i \leq \alpha'_{i+1}$. For every x we can now easily find a *left* and *right* subvector containing only smaller and equal, or greater values. $(\alpha'_1, \alpha'_2, \ldots, \alpha'_n)$ corresponds to a discrete $(k+1)$-state Markov chain, where each state corresponds to observing a particular sample, and state k corresponds to the observation of the largest observed sample.

For this Markov chain, the complementary cumulative distribution function (CCDF) can be found as

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

$G(x)$ in Equation 1 corresponds to the probability of being in any state that corresponds to the random variable X taking on a value larger than x, which corresponds to the CCDF, as shown in Figure 1.

3.3 Markov Chains

For every position x in the $(k+1)$-state Markov chain, a 2-state Markov chain as in Figure 2 can be obtained, where the first state corresponds to the random variable X taking on values $X \leq x$ and the second state to $X > x$.

The transition probabilities $p_0(x)$, $p_1(x)$ can be found as follows:

$$\begin{aligned} p_0(x) &= \frac{1}{F(x)} \sum_{r=0}^{i-1} P_r \sum_{j=i}^{k} p_{rj} \quad \text{for } i-1 \leq x < i,\ i = 1, 2, \ldots, k \\ p_1(x) &= \frac{1}{G(x)} \sum_{r=i}^{k} P_r \sum_{j=0}^{i-1} p_{rj} \quad \text{for } i-1 \leq x < i,\ i = 1, 2, \ldots, k \end{aligned} \tag{2}$$

3.4 Local Correlation Coefficient

The covariance normalized by the standard deviation is the *correlation coefficient* ρ:

$$\rho_{XY} = Corr(X, Y) = \frac{Cov(X, Y)}{\sigma_X \sigma_Y} \tag{3}$$

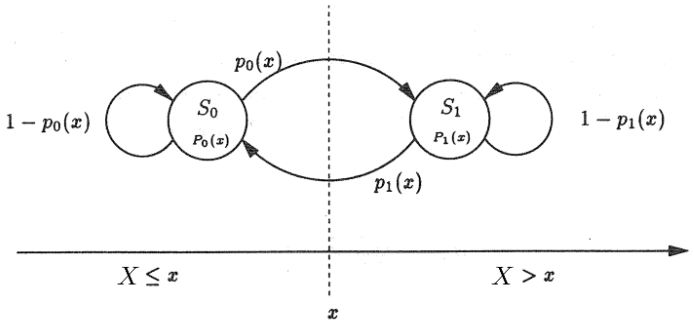


Figure 2: A local x-based 2-state Markov chain obtained from a $(k+1)$-state Markov chain. From [5].

where $-1 \leq \rho \leq 1$.

While ρ considers all values of the random variables X, Y, the *local* correlation coefficient $\rho(x)$ is a function of x, and therefore depends on this position x. Considering 2-state Markov chains, their global correlation coefficient ρ corresponds to the local correlation coefficient $\rho(x)$ of the underlying $(k+1)$-state Markov chain at the respective position x.

The local coefficient can be found as $\rho(x) = 1 - [p_0(x) + p_1(x)]$, where for $p_0(x) + p_1(x) < 1$ a positive correlation is found according to [5]. In other words, this means that there is a larger probability that once either state is entered, the process will remain in that state for one or more iterations, than if the state has not been entered. This corresponds to *burstiness* of for example packet delays, where it is likely to observe more large delays once a single large delay has been observed. With this, the state probabilities $P_0(x)$, $P_1(x)$ for the 2-state Markov chain of being in state $S_0(x)$, $S_1(x)$ respectively are

$$\begin{aligned} P_0(x) &= \frac{p_1(x)}{1 - \rho(x)} \\ P_1(x) &= 1 - P_0(x) = \frac{p_0(x)}{1 - \rho(x)} \end{aligned} \tag{4}$$

3.5 Procedure

The algorithm aims to determine – through simulation – the CCDF $G(x)$ of a $(k+1)$-state Markov chain where k is given a-priori, and where transition probabilities p_{ji} are *not* known.

To do this, we count how many times each state i has been entered after n transitions in the chain. We save this number in the counter variable h_i, from which we can find the *state frequency*

$$v_i = \sum_{j=i}^{k} h_j \quad \text{for } i = 0,\, 1,\, \ldots,\, k \quad \text{with } v_0 = n \tag{5}$$

that tells us how many times the right state $S_1(x)$ with $i - 1 \leq x < i$ of the 2-state Markov chain has been entered – for reference see Figure 2. $r_i = n - v_i$ handles the left state $S_0(x)$ and is implicitly counted.

The *transition frequency* c_i, $i = 1, 2, \ldots, k$ counts how many times the transition $S_1(x) \rightarrow S_0(x)$ across the dividing line of x has happened. Frequencies v_i and c_i and the corresponding

counterparts $r_i = n - v_i$ and $a_i \approx c_i$ belong to the equivalent x-based 2-state Markov chain. An overview is given in Figure 3.

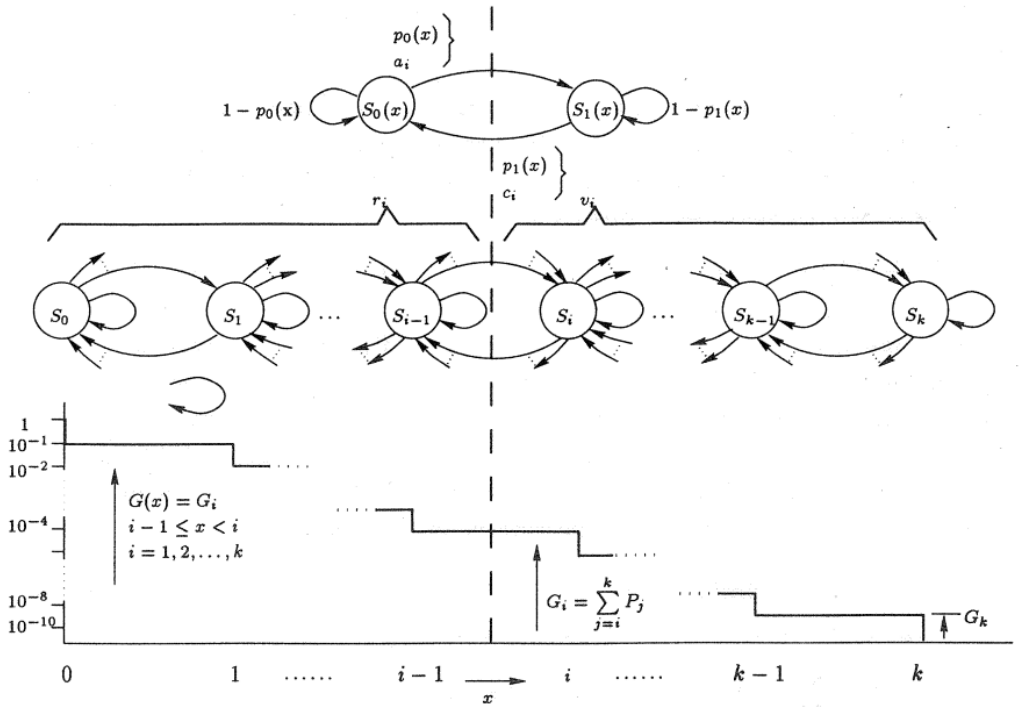


Figure 3: An overview of the transformation from a $(k+1)$-state to a 2-state Markov chain, as well as counters a_i, c_i, r_i, v_i and the association with G_i. From [5].

Next, after having measured v_i, c_i from n samples, the *large sample conditions* are checked:

$n \ge 10^3$ require "many" samples; makes sure the transient phase has been left
$(r_i,\, v_i) \ge 10^2$ at least 100 samples left and right of state i
$c_i,\, a_i \ge 10$ at least 10 transitions from state i
$v_i - c_i \ge 10$ have at least 10 more visits than leaves (sometimes remain in state)

and if met, we can determine (a) the measured CCDF $\widetilde{G}(x) = \widetilde{G}_i = \frac{v_i}{n}$, (b) the measured state average $\widetilde{\alpha} = \frac{1}{n}\sum_{i=1}^{k} v_i$, (c) the measured local correlation coefficient $\widetilde{\rho}(x) = \widetilde{\rho}_i = 1 - \frac{c_i/v_i}{1-v_i/n}$, (d) the corresponding correlation factor $\widetilde{cf}(x) = \widetilde{cf}_i = \frac{1+\widetilde{\rho}_i}{1-\widetilde{\rho}_i}$ and (e) the relative error $d_G(x) = d_i = \sigma_G(x)/\widetilde{G}(x) = \sqrt{\left(\frac{1-v_i/n}{v_i} \cdot \widetilde{cf}_i\right)}$.

Here $\sigma_G(x)$ is the standard deviation of $P_1(x)$ in the 2-state Markov chain, which was found in [7] and [8] to be normally distributed. As the transitions are counted in c_i, not only the local correlation coefficient $\widetilde{\rho}(x)$ but also the relative error $d_G(x)$ as a function of $\widetilde{\rho}(x)$ can be found. This allows the algorithm to continuously increase the number of trials n until the

error condition $d_i \leq d_{max}$ is met, where d_{max} is an a-priori maximum error. The standard deviation $\sigma_G(x)$ gives an absolute error, which is made relative to the probability of being in state $\geq i$ of the Markov chain by dividing with the value of the CCDF at the respective position $\widetilde{G}(x)$. The correlation coefficient becomes large for a large, positive correlation, and so increases the relative error. A large correlation corresponds to many transitions from a state to itself (e.g. burst errors resulting in a long sojourn time in the large-error-state), and so the algorithm demands more observations in this case to ensure that the transition probabilities *between* states are accurately modelled.

3.6 Executing the algorithm

Parameters for a simulation supervised by LRE are d_{max}, x_{min}, x_{max}, x_{step}. The sequence of samples x_1, x_2, …, x_n is mapped to the $(k+1)$-state Markov chain, and n is increased until $d_i \leq d_{max}$ for all i. At runtime an index s corresponds to that state in the Markov chain whose error d_s shall be checked next, and it is initialized at $s = 1$.

Algorithm 1 LRE Algorithm; adapted from [5]

```
1:  procedure LRE_SIMULATION(maximum error d_max, bounds of evaluation range x_min, x_max,
    interval size x_step)
2:      k = (x_max − x_min)/x_step                          ▷ largest state index
3:      for i = 0, …, k do
4:          h_i = 0; c_i = 0                                ▷ initialize counters
5:      x = 0; n = 0; s = 1
6:      while s ≠ k do                                      ▷ iterate through states 1 … k
7:          ω = x                                           ▷ remember old value
8:          generate new observation x
9:          increment counter h_x
10:         increment number of samples n
11:         if x < ω then                                   ▷ transition to left of ω
12:             for i = x + 1 to ω do
13:                 increment c_i                           ▷ state transition to smaller value
14:         calculate ρ̃_s and relative error d_s
15:         if d_s ≤ d_max then                             ▷ target error reached?
16:             increment s                                 ▷ go to next state
17:     for i = 1, …, k do
18:         calculate sum frequencies v_i from Equation 5
19:         calculate result values G̃_i, ρ̃_i, d_i
```

4 Application

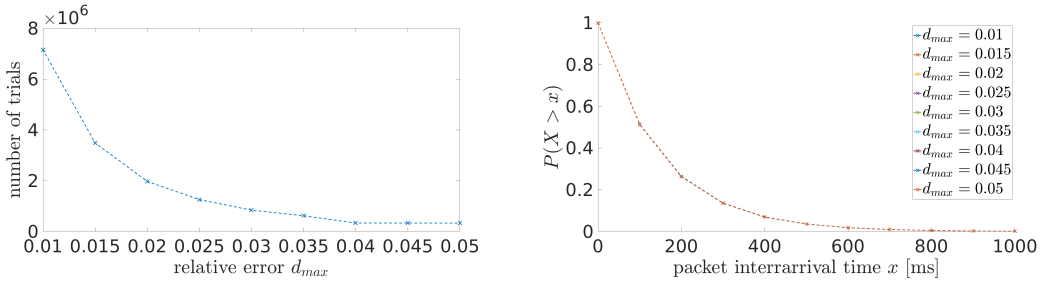
A standalone LRE implementation is available on GitHub in [9], which is also used for our OMNeT++ integration available in [10]. The novel `LRE` OMNeT++ entity subscribes to a `signal` specified by the user. When enough signal emissions were captured and `LRE` deems results confident enough, the simulation is ended. The intended usage therefore foresees setting no simulation time, but configuring the `LRE` entity so that it can take over controlling the simulation time.

Configuration of the entity is done through setting NED parameters. The output corresponds to the measured probability of the CCDF at all x-positions, as well as the relative error $d(x)$, local correlation coefficient $\rho(x)$, standard deviation $\sigma(x)$, number of samples and transitions per state $n(x)$, $t(x)$.

4.1 Example

An example use case is included in the implementation in [10]. A simple network with one sender and one receiver is modelled. The sender draws the time it waits until sending the next packet from an exponential distribution with a mean of 150 ms. We quantise the packet interarrival times measured at the receiver into intervals of size $x_{step} = 100\,\text{ms}$ and observe the range $x_{min} = 0\,\text{ms} \leq x \leq x_{max} = 1000\,\text{ms}$. We repeat the LRE runs for the maximum errors $d_{max} \in [0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04, 0.045, 0.05]$ to see how the target error affects the required number of observations.

This simple approach is chosen so that it is easily understood and imitated by new users of the LRE method. As all code required to replicate the exact results is part of the open-source code release in [10], it shall provide easy access to understanding how the LRE method is used in OMNeT++.



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

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

Figure 4: Number of observations requested and CCDF measured by the LRE for OMNeT++ integration of packet interarrival times measured at the receiver.

Figure 4a shows that the number of observations increases strongly as the maximum allowed error d_{max} decreases. In our test case, when $d_{max} = 1\,\%$ then the number of observations is 27 times larger than the number of observations for $d_{max} = 5\,\%$. Figure 4b shows the corresponding CCDF, where the three cases of d_{max} correspond to measured probabilities that differ in the range of 0.1 %. The largest packet interrarival times occur at probability $P(X > 1000\,\text{ms}) \approx 10^{-3}$.

We can conclude that the LRE algorithm successfully controls the number of observations it requires and so the simulation time. When a more rigid analysis is required, d_{max} is set to a stricter limit. In our simple scenario the difference in the CCDF is marginal. When a different range of the statistic in question is required, then this is also easily configured through the algorithm input.

5 Summary, Conclusion, and Outlook

In this paper we have demonstrated a lesser-known but useful statistical method for result evaluation that is capable of deciding when a simulation shall end automatically, ensuring that an a-priori target confidence level is met. The LRE algorithm can be configured so that the intended resolution of some statistic is measured – so if the user is interested in rare events, then the target resolution is easily configured, and the algorithm will ensure that a confident statement can be made afterwards.

We have provided a new description of the LRE algorithm which dates back to the eighties and nineties. An OMNeT++ network module is implemented that provides the algorithm functionality to the simulator. Through simple network configuration the algorithm is configured, and when the simulation time is unset or set to a very large value, the LRE entity will handle simulation termination when the results have reached the target confidence level. This allows the easy analysis of arbitrarily rare event statistics in OMNeT++. Events that occur with a very small probability can not be investigated more quickly as the simulation time may still be prohibitively long; so far the gain is the automatic on-line evaluation and termination when the target confidence is met. The LRE method was combined in [11] with the RESTART method, first published in [12], to significantly decrease the required simulation time. An implementation of this in OMNeT++ could prove useful to many researchers.

We have made available both the standalone implemenation in [9] as well as the OMNeT++ integration in [10]. The network model that generates the data shown in this paper is part of the release in [10].

References

[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] F. Schreiber, "Time Efficient Simulation The LRE-Algorithm for Producing Empirical Distribution Functions with Limited Relative Error," *AEÜ - International Journal of Electronics and Communications*, vol. 2, no. 38, 1984.

[4] ——, "Effective Control of Simulation Runs by a New Evaluation Algorithm for Correlated Random Sequences," *AEÜ - International Journal of Electronics and Communications*, 1988.

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

[6] K. Below, L. Battaglia, and U. Killat, "RESTART/LRE Simulation - The Reliability Issue," Hamburg University of Technology, Tech. Rep., 1999.

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

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

[9] "LRE Implementation." [Online]. Available: https://doi.org/10.5281/zenodo.1312970

[10] "LRE OMNeT++ Integration." [Online]. Available: https://doi.org/10.5281/zenodo.1313054

[11] 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.

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