Square Grid Benchmarks for *Source-Terminal* Network Reliability Estimation*

Roger Paredes† Leonardo Duenas-Osorio

Rice University, Houston TX, USA.
03/2018

This document describes a synthetic benchmark data set that can be used to compare network reliability estimation methods [1, 2]. The document is structured as follows: Section 1 introduces the *source-terminal* network reliability problem and an input file format used to encode instances of the problem. Section 2 provides details on the set of benchmarks we consider and the setup for numerical experiments. Section 3 gives reference computations via exact bounds and crude Monte Carlo simulation estimates. Section 4 concludes this benchmarking study.

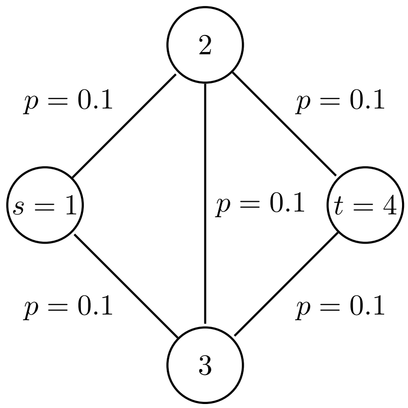
1 Introduction

The *source-terminal* network reliability problem we consider is defined as follows. Given an undirected graph $G(V, E)$, where V is the set of nodes and E is the set of links, the *source-terminal* network reliability problem asks to compute the probability that nodes $s, t \in V$ remain connected, here denoted $r_{s,t}$, given that every link $e \in E$ fails with known probability p_e.

Physical networks studied in practice tend to be highly reliable, thus the real challenge is that of knowing the failure probability of the network (or unreliability), herein denoted $u_{s,t}$. In the exact estimation setting $u_{s,t} = 1 - r_{s,t}$. However, in the approximation setting we focus on computing $\hat{u}_{s,t}$ directly with small relative error $|\hat{u}_{s,t} - u_{s,t}|/u_{s,t}$, otherwise an estimate of $r_{s,t}$ when $r_{s,t} \approx 1$ would give a poor estimate of $u_{s,t}$ in terms of relative error.

*cite as Paredes and Duenas-Osorio [1] or Paredes et al. [2].

†roger.paredes(at)rice(dot)edu




```
"benchmarks/example.txt"
c This c line is a comment. The next T
c line contains s-t nodes.
T 1 4
c The next e lines list edges and
c their respective reliabilities.
e 1 2 0.90
e 1 3 0.90
e 2 3 0.90
e 2 4 0.90
e 3 4 0.90
c This is the end of the file.
```

Figure 1: Left: An example instance of the *source-terminal* network reliability problem. Right: Associated input format as shown in the contents of the file "example.txt".

To showcase computations for the *source-terminal* network reliability problem, we describe next the file format to represent an instance of the problem. As an example, the file located in "`benchmarks/example.txt`" contains a description of the instance shown in Fig. 1. In particular, every line is interpreted as follows:

- Lines starting with a "c" character contain comments that are ignored when parsing the input into computer programs.
- A line starting with a "T" character contains the set of terminals. In this case it contains nodes s and t in V. Thus, the "T" character is followed by s and t node indices.
- Lines starting with an "e" character contain edge information. Thus, for an edge $(i, j) = e \in E$ such that $i, j \in V$, the character "e" is followed by node indices i and j as well as its respective reliability $1 - p_e$.

The set of benchmarks contained in the folder "`benchmarks`" uses the instance format just described. In the following section we describe these benchmarks in detail.

2 The Square Grid Benchmarks

We consider square grid networks of size $N \times N$. While there are numerous benchmark possibilities, grids do offer the following advantages:

- For $N > 2$ they cannot be reduced using parallel-series techniques.
- Their size can be arbitrarily increased until rendering exact methods impractical.
- Element failure probabilities can be varied to challenge simulation methods in the rare-event regime.

In this document we consider uniform edge failure probabilities p and let the source-terminal nodes be a pair of nodes located at opposite corners of the $N \times N$ square grid (e.g. Fig. 2). Also, we consider various values of the size parameter N and various values for the failure probability p. In particular, we consider:

- $N = 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, \ldots, 100$.
- 5 cases of the failure probability $p = 1 \times 10^{-p_r}$, where p_r is the rarity parameter and we let $p_r = 1, 2, 3, 4, 5$.

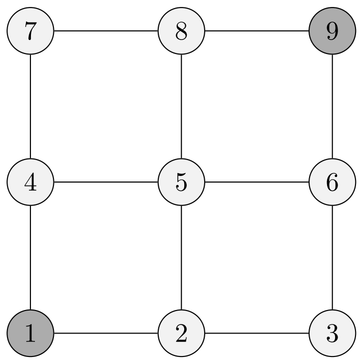
The folder "`benchmarks`" contains subfolders "`benchmarks/p`$\{p_r\}$", and each subfolder contains input instances "`benchmarks/p`$\{p_r\}$`/grid`$\{N\}$`.txt`". For example, the grid network of size 100×100 and edge failure probabilities of $p = 1 \times 10^{-5}$ is located in "`benchmarks/p5/grid100.txt`".

As a reference, we provide preliminary reliability estimates for this set of benchmarks using crude Monte Carlo simulation (CMC) as well as methods that have been popularized in the Civil Engineering community. More specifically, we use the State Space Partition (SSP) method [1], a generalization of the Selective Recursive Decomposition [3], to estimate theoretical bounds on the failure probability of networks.

We used a high-performance cluster to run numerical experiments in parallel. Each node of the cluster had a 12-core 2.83 GHz processor, with 48GB of main memory (4GB each core), and each experiment was run on a single core. Each experiment consisted of running a method (CMC or SSP) on a given benchmark. The timeout for each experiment was set to 7.8 hours.

The following section discusses experiment results, while highlighting main trends and observations.

$p = 1 \times 10^{-1}$ for all edges.



"benchmarks/p1/grid3.txt"

```
c Terminals
T 1 9
c edges and up probabilities
e 1 2 0.9000000000
e 1 4 0.9000000000
e 2 3 0.9000000000
e 2 5 0.9000000000
e 3 6 0.9000000000
e 4 5 0.9000000000
e 4 7 0.9000000000
e 5 6 0.9000000000
e 5 8 0.9000000000
e 6 9 0.9000000000
e 7 8 0.9000000000
e 8 9 0.9000000000
```

Figure 2: Grid Network of size 3×3 (i.e. $N = 3$). Source and terminal nodes are the ones shaded at the corners.

3 Numerical Experiments

We provide tabulated results summarizing the experiment outcomes. For Crude Monte Carlo simulation (CMC) we display the unreliability estimate $\hat{u}_{s,t}$, sample variance $\hat{\sigma}^2$, the number of sample draws N_s we obtained within the timeout threshold of 7.8 hours, and the estimator coefficient of variation CoV. Tables 1-5 summarize results for all cases of the size parameter N and edge failure probabilities p. It is easy to see that, as the failure probability becomes smaller, the estimators become less reliable. For example, we can use our knowledge of the exact coefficient of variation squared for Bernoulli trials and realize that $\sigma^2/\mu^2 = 1/u_{s,t} - 1$. Thus, the smaller the exact failure probability $u_{s,t}$, the greater the error in CMC experiments.

In practice, for a finite yet large sample size N_s, the central limit theorem is assumed valid such that confidence intervals can be obtained from the sample mean $\hat{u}_{s,t}$ and the sample variance $\hat{\sigma}^2$. For example, $\hat{u}_{s,t} \pm 1.96 \cdot \hat{\sigma}/\sqrt{N_s}$ would give the widely used 95% confidence intervals. Clearly, CMC is going to be challenged by small values of $u_{s,t}$ and its rate of convergence. Thus, when possible, a more reliable approach consists of computing bounds on $u_{s,t}$ with 100% confidence.

For the theoretical method, namely the State Space Partition (SSP) method, we display lower and upper bounds, denoted "$\leq u_{s,t}$" and "$\geq u_{s,t}$", respectively. Also, we show the runtime in seconds. Tables 6-10 summarize

Table 1: CMC estimates for $p = 1 \times 10^{-1}$.

N	$\hat{u}_{s,t}$	$\hat{\sigma}^2$	N_s	CoV	Time [s]
3	0.027495	0.026739	360528500	0.000313	28080.00
4	0.024968	0.024344	230806400	0.000411	28080.00
5	0.024433	0.023836	159286900	0.000501	28080.00
6	0.024349	0.023756	116834700	0.000586	28080.00
7	0.024302	0.023711	88625840	0.000673	28080.00
8	0.024346	0.023754	68984150	0.000762	28080.00
9	0.024350	0.023757	46792120	0.000925	28080.00
10	0.024367	0.023773	43924870	0.000955	28080.00
20	0.024321	0.023729	11250640	0.001888	28080.00
30	0.024316	0.023725	4986994	0.002837	28080.00
40	0.024276	0.023687	2758273	0.003817	28080.00
50	0.024197	0.023611	1636627	0.004964	28080.00
60	0.024277	0.023688	1110440	0.006016	28080.02
70	0.024292	0.023702	777959	0.007185	28080.00
80	0.024197	0.023612	605777	0.008159	28080.02
90	0.024524	0.023923	468106	0.009218	28080.03
100	0.024625	0.024019	375264	0.010274	28080.00

results for all benchmark cases. These bounds refer to the "best bounds" as defined in Paredes and Duenas-Osorio [1]. Clearly, the size parameter N is the most sensitive since bounds tend to be wider as N increases. Also, when bounds converge the failure probability has no effect. However, when the bounds do not converge, both the size and rarity of failure probabilities have an effect on how tight the resulting bounds are. More specifically, for smaller failure probabilities bounds tend to be tighter. Also, we highlight that best bounds obtained by the method introduced in Paredes and Duenas-Osorio [1] give consistently practical lower bounds, even for $N = 100$. Nevertheless, the upper bound becomes unpractical as the size parameter increases.

For convenience, results in Tables 1-10 are saved in readable text using the comma separated values (CSV) format. For example, for a fixed value of $p_r = 1$, tabulated results for CMC as in Table 1 can be found under "`logs/log_cmc_p1.txt`".

4 Concluding Remarks

Network reliability estimation continues to be untractable in practice, as exact methods and simulation methods remain challenged by both the size

Table 2: CMC estimates for $p = 1 \times 10^{-2}$.

N	$\hat{u}_{s,t}$	$\hat{\sigma}^2$	N_s	CoV	Time [s]
3	0.000207	0.000207	385711900	0.003539	28080.00
4	0.000204	0.000204	249361400	0.004430	28080.00
5	0.000204	0.000204	170186400	0.005369	28080.00
6	0.000206	0.000206	124414700	0.006248	28080.00
7	0.000204	0.000204	92643560	0.007281	28080.00
8	0.000206	0.000206	73038950	0.008143	28080.00
9	0.000200	0.000200	58639050	0.009229	28080.00
10	0.000204	0.000204	39187910	0.011174	28080.00
20	0.000211	0.000211	10122320	0.021655	28080.00
30	0.000201	0.000201	4397413	0.033592	28080.00
40	0.000198	0.000198	2325853	0.046621	28080.00
50	0.000207	0.000207	1538490	0.056071	28080.01
60	0.000205	0.000205	1016046	0.069330	28080.02
70	0.000202	0.000201	729454	0.082470	28080.02
80	0.000201	0.000201	562967	0.094063	28080.04
90	0.000196	0.000196	432778	0.108455	28080.04
100	0.000168	0.000168	445975	0.115460	28080.02

Table 3: CMC estimates for $p = 1 \times 10^{-3}$.

N	$\hat{u}_{s,t}$	$\hat{\sigma}^2$	N_s	CoV	Time [s]
3	1.992048e-06	1.992044e-06	394066800	0.035691	28080.00
4	2.161132e-06	2.161127e-06	254033500	0.042679	28080.00
5	1.972440e-06	1.972436e-06	170347400	0.054554	28080.00
6	1.747683e-06	1.747680e-06	125880900	0.067420	28080.00
7	1.971131e-06	1.971127e-06	95376730	0.072932	28080.00
8	2.041193e-06	2.041189e-06	74956150	0.080845	28080.00
9	1.868513e-06	1.868510e-06	58870340	0.095346	28080.00
10	1.718392e-06	1.718389e-06	47719040	0.110431	28080.00
20	2.070359e-06	2.070355e-06	12075200	0.200000	28080.00
30	1.483843e-06	1.483841e-06	5391407	0.353553	28080.00
40	1.808447e-06	1.808444e-06	2764803	0.447213	28080.00
50	1.111608e-06	1.111608e-06	1799195	0.707107	28080.00
60	8.220730e-07	8.220730e-07	1216437	1.000000	28080.00
70	0.000000e+00	0.000000e+00	865940	NaN	28080.02
80	3.030152e-06	3.030147e-06	660033	0.707106	28080.03
90	1.916770e-06	1.916770e-06	521711	1.000000	28080.04
100	0.000000e+00	0.000000e+00	454785	NaN	28080.02

Table 4: CMC estimates for $p = 1 \times 10^{-4}$.

N	$\hat{u}_{s,t}$	$\hat{\sigma}^2$	N_s	CoV	Time [s]
3	7.655205e-09	7.655205e-09	391890200	0.577350	28080.00
4	2.379872e-08	2.379872e-08	252114400	0.408248	28080.00
5	1.161015e-08	1.161015e-08	172263100	0.707107	28080.00
6	2.349814e-08	2.349814e-08	127669700	0.577350	28080.00
7	3.818167e-08	3.818167e-08	78571740	0.577350	28080.00
8	3.222889e-08	3.222889e-08	62056120	0.707107	28080.00
9	0.000000e+00	0.000000e+00	49193670	NaN	28080.00
10	0.000000e+00	0.000000e+00	41207310	NaN	28080.00
20	0.000000e+00	0.000000e+00	9946345	NaN	28080.00
30	0.000000e+00	0.000000e+00	4459871	NaN	28080.00
40	0.000000e+00	0.000000e+00	2431569	NaN	28080.00
50	0.000000e+00	0.000000e+00	1531661	NaN	28080.00
60	0.000000e+00	0.000000e+00	1014298	NaN	28080.02
70	0.000000e+00	0.000000e+00	751584	NaN	28080.02
80	0.000000e+00	0.000000e+00	558672	NaN	28080.03
90	0.000000e+00	0.000000e+00	448258	NaN	28080.02
100	0.000000e+00	0.000000e+00	358041	NaN	28080.06

Table 5: CMC estimates for $p = 1 \times 10^{-5}$.

N	$\hat{u}_{s,t}$	$\hat{\sigma}^2$	N_s	CoV	Time [s]
3	0.000000e+00	0.000000e+00	322488500	NaN	28080.00
4	0.000000e+00	0.000000e+00	212580500	NaN	28080.00
5	0.000000e+00	0.000000e+00	144023700	NaN	28080.00
6	0.000000e+00	0.000000e+00	103975100	NaN	28080.00
7	0.000000e+00	0.000000e+00	78667840	NaN	28080.00
8	0.000000e+00	0.000000e+00	62030430	NaN	28080.00
9	1.966564e-08	1.966564e-08	50850120	1.0	28080.00
10	0.000000e+00	0.000000e+00	40639410	NaN	28080.00
20	0.000000e+00	0.000000e+00	10186810	NaN	28080.00
30	0.000000e+00	0.000000e+00	4435675	NaN	28080.00
40	0.000000e+00	0.000000e+00	2499222	NaN	28080.00
50	0.000000e+00	0.000000e+00	1533004	NaN	28080.00
60	0.000000e+00	0.000000e+00	1045440	NaN	28080.00
70	0.000000e+00	0.000000e+00	757264	NaN	28080.03
80	0.000000e+00	0.000000e+00	572322	NaN	28080.00
90	0.000000e+00	0.000000e+00	446586	NaN	28080.02
100	0.000000e+00	0.000000e+00	363252	NaN	28080.02

Table 6: SSP estimates for $p = 1 \times 10^{-1}$.

N	$\le u_{s,t}$	$\ge u_{s,t}$	Time [s]
3	2.7497828593e-02	2.7497828593e-02	1.22
4	2.4953650423e-02	2.4953650423e-02	206.49
5	2.4443410495e-02	2.4443410495e-02	17845.83
6	2.4326683271e-02	1.1545304605e-01	28080.25
7	2.4320145065e-02	1.1814674212e-01	28080.08
8	2.4282119282e-02	1.1980542156e-01	28080.79
9	2.4331994910e-02	1.1980542156e-01	28081.26
10	2.4319970641e-02	5.5228597094e-01	28081.17
20	2.4285225883e-02	9.9127294031e-01	28080.32
30	2.4287575021e-02	9.9900193336e-01	28080.89
40	2.4282064396e-02	1.0000000000e+00	28080.47
50	2.4282063212e-02	1.0000000000e+00	28080.28
60	2.4282050156e-02	1.0000000000e+00	28080.35
70	2.4281864714e-02	1.0000000000e+00	28080.25
80	2.4280866453e-02	1.0000000000e+00	28080.18
90	2.4275485956e-02	1.0000000000e+00	28080.38
100	2.4241322212e-02	1.0000000000e+00	28080.12

Table 7: SSP estimates for $p = 1 \times 10^{-2}$.

N	$\le u_{s,t}$	$\ge u_{s,t}$	Time [s]
3	2.0798597679e-04	2.0798597679e-04	1.20
4	2.0409116253e-04	2.0409116253e-04	203.02
5	2.0403122638e-04	2.0403122638e-04	17859.22
6	2.0403040483e-04	2.4201074932e-04	28080.19
7	2.0403039597e-04	2.3221412260e-04	28080.17
8	2.0403034040e-04	2.0185072696e-04	28080.68
9	2.0403039799e-04	2.0858740071e-04	28081.37
10	2.0403034040e-04	5.4793964334e-03	28080.60
20	2.0402897438e-04	1.3480095400e-01	28081.16
30	2.0402897438e-04	3.3460268034e-01	28080.85
40	2.0397898055e-04	7.2035636932e-01	28080.42
50	2.0397898055e-04	8.5661128560e-01	28080.33
60	2.0397898055e-04	9.2669213096e-01	28080.28
70	2.0397898055e-04	9.5336259770e-01	28080.17
80	2.0397898055e-04	9.7032996155e-01	28080.40
90	2.0397898055e-04	9.8112435217e-01	28080.17
100	2.0397898055e-04	9.8799158681e-01	28081.19

Table 8: SSP estimates for $p = 1 \times 10^{-3}$.

N	$\leq u_{s,t}$	$\geq u_{s,t}$	Time [s]
3	2.0079989600e-06	2.0079989600e-06	1.25
4	2.0040090120e-06	2.0040090120e-06	204.14
5	2.0040030120e-06	2.0040030120e-06	17772.96
6	2.0040079128e-06	2.0071351646e-06	28080.17
7	2.0040030040e-06	2.0071355646e-06	28080.15
8	2.0040030039e-06	4.6752864837e-06	28080.89
9	2.0040030039e-06	2.3251507015e-06	28080.54
10	2.0040030039e-06	8.7338433202e-06	28080.65
20	2.0040029900e-06	2.0816183241e-03	28081.00
30	2.0040029900e-06	6.7570300814e-03	28080.75
40	2.0039979900e-06	1.3041116971e-01	28080.46
50	2.0039979900e-06	1.8633581325e-01	28080.29
60	2.0039979900e-06	2.2904871125e-01	28080.25
70	2.0039979900e-06	2.6298910334e-01	28080.32
80	2.0039979900e-06	2.9543530217e-01	28080.22
90	2.0039979900e-06	3.2645308817e-01	28080.46
100	2.0039979900e-06	3.5610534585e-01	28080.15

Table 9: SSP estimates for $p = 1 \times 10^{-4}$.

N	$\leq u_{s,t}$	$\geq u_{s,t}$	Time [s]
3	2.0007999900e-08	2.0007999900e-08	1.25
4	2.0004000900e-08	2.0004000900e-08	202.35
5	2.0004000300e-08	2.0004000300e-08	17818.31
6	2.0004000300e-08	2.0004433478e-08	28080.18
7	2.0004000300e-08	2.0004315610e-08	28080.15
8	2.0004000300e-08	2.2630750518e-08	28080.68
9	2.0004000300e-08	2.0004000661e-08	28081.42
10	2.0004000300e-08	2.0354197423e-08	28081.19
20	2.0004000300e-08	3.0486183852e-05	28080.32
30	2.0004000300e-08	7.2809834903e-05	28080.84
40	2.0003999800e-08	1.3988733361e-02	28080.44
50	2.0003999800e-08	2.0479851949e-02	28080.31
60	2.0003999800e-08	2.5666177110e-02	28080.51
70	2.0003999800e-08	3.0041047220e-02	28080.15
80	2.0003999800e-08	3.4396273663e-02	28080.14
90	2.0003999800e-08	3.8731944642e-02	28080.82
100	2.0003999800e-08	4.3048147964e-02	28080.10

Table 10: SSP estimates for $p = 1 \times 10^{-5}$.

N	$\leq u_{s,t}$	$\geq u_{s,t}$	Time [s]
3	2.0000799999e-10	2.0000799999e-10	1.22
4	2.0000400009e-10	2.0000400009e-10	204.98
5	2.0000400003e-10	2.0000408862e-10	17795.58
6	2.0000400003e-10	2.0000406367e-10	28080.17
7	2.0000400003e-10	2.0000383927e-10	28080.14
8	2.0000400003e-10	2.0268150995e-10	28080.74
9	2.0000400003e-10	2.0000402850e-10	28081.38
10	2.0000400003e-10	2.0042762846e-10	28081.18
20	2.0000400003e-10	3.0633573682e-07	28080.26
30	2.0000400003e-10	7.3358723053e-07	28080.69
40	2.0000399998e-10	1.4088817258e-03	28080.43
50	2.0000399998e-10	2.0677842896e-03	28080.33
60	2.0000399998e-10	2.5966358937e-03	28080.21
70	2.0000399998e-10	3.0453686788e-03	28080.18
80	2.0000399998e-10	3.4938995785e-03	28080.31
90	2.0000399998e-10	3.9422286838e-03	28082.31
100	2.0000399998e-10	4.3903560853e-03	28081.76

of problem instances and the rarity of system failures, respectively. Perhaps a case-by-case approach will yield the best results in practice.

The present document is subject to changes to better reflect ongoing efforts on system reliability estimation by the SISSRA research group.

5 Acknowledgements

The input file format is adopted from related work by Professor Radislav Vaisman[1]. Also, we implement reliability estimation algorithms in Python leveraging the network analysis module NetworkX [4]. Furthermore, in regards to computational resources used in our numerical experiments, we thank the Data Analysis and Visualization Cyberinfrastructure funded by NSF under grant OCI-0959097 and Rice University. Finally, the authors gratefully acknowledge the support by the U.S. National Science Foundation (Grants CMMI-1436845 and CMMI-1541033).

[1] `https://people.smp.uq.edu.au/RadislavVaisman/`

References

[1] Roger Paredes, Leonardo Duenas-Osorio, and Isaac Hernandez-Fajardo. Decomposition Algorithms for Reliability Estimation of Interdependent Lifeline Systems. *Earthquake Engineering & Structural Dynamics (accepted)*, 2018. doi: 10.1002/eqe.3071.

[2] Roger Paredes, Leonardo Duenas-Osorio, Kuldeep S. Meel, and Moshe Y. Vardi. Network Reliability Estimation in Theory and Practice. *Reliability Engineering & System Safety (submitted)*, jun 2018. URL `http://arxiv.org/abs/1806.00917`.

[3] Hyun-woo Lim and Junho Song. Efficient risk assessment of lifeline networks under spatially correlated ground motions using selective recursive decomposition algorithm. *Earthquake Engineering & Structural Dynamics*, 41(13):1861–1882, oct 2012.

[4] Daniel A. Schult. Exploring network structure, dynamics, and function using networkx. In *In Proceedings of the 7th Python in Science Conference (SciPy)*, pages 11–15, 2008.