13th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP13
Seoul, South Korea, May 26-30, 2019

A Weighted Model Counting Approach for Critical Infrastructure Reliability

Roger Paredes
Graduate Student, Dept. of Civil and Environmental Engineering, Rice University, Houston, Texas, United States

Leonardo Dueñas-Osorio
Professor, Dept. of Civil and Environmental Engineering, Rice University, Houston, Texas, United States

Kuldeep S. Meel
Assistant Professor, Computer Science Department, National University of Singapore, Singapore

Moshe Y. Vardi
Professor, Dept. of Computer Science, Rice University, Houston, Texas, United States

ABSTRACT: Reliability assessment of engineered systems such as telecommunication networks, power grids, and railroads is an important step towars supporting resilient communities. However, calculating the reliability of a network is computationally intensive. Thus, simulation methods are often preferred over exact methods in practice. Unfortunately, highly reliable and large scale systems can challenge common assumptions in simulation techniques, rendering reliability estimates—as well as reported error and confidence—unreliable themselves. A new generation of techniques, termed *probably approximately correct* (PAC) methods, delivers *provable* network reliability calculations with user-specified error and confidence. In this paper we focus on RelNet, a model counting-based method for network reliability estimation endowed with rigorous PAC guarantees. Despite previous success in power transmission network applications, small edge failure probabilities and dependent failures can challenge the current methodology. We put forward Weighted RelNet, a general importance sampling-based extension that treats the system's joint probability distribution as a black box. Empirical evaluations suggest the new approach is competitive across challenging rare-event benchmarks.

1. Introduction

Critical infrastructures such as power grids, water delivery, and gas supply systems provide goods an services on which modern societies thrive. Given their importance and vulnerability, the engineering concern of how reliable lifeline systems are, arises naturally in our pursuit of resilient communities.

A common surrogate model to study critical infrastructure consists of a stochastic graph consistent with a given network topology and component fragilities. In this paper, we consider the K-terminal network reliability problem (Ball et al., 1995), which takes as input a graph $G(V,K,E)$ and edge failure probabilities $P=(p_e)_{e\in E}$. Here V is the set of vertices, $K \subseteq V$ is the set of terminals, E is the set of edges, and every edge $e \in E$ fails with arbitrary probability p_e. We assume that vertices are perfectly reliable without loss of generality due to well known reductions (Ball et al., 1995, Section 2.2). The K-terminal network reliability problem asks to compute $r_G(P)$, the probability that vertices in K are connected. The problem belongs to the #P–complete complexity class, thus largely believed to be computationally intractable. As a consequence, for general graphs, past and ongoing efforts focus on approximations and Monte Carlo-based approaches.

In the practical non-exact setting, we care about quantifying unreliability $u_G(P) = 1 - r_G(P)$ with acceptable relative error, denoted ε,[1] because in engineering practice, systems are highly reliable by design—i.e. $u_G(P) \to 0$ and $r_G(P) \to 1$. Despite previous success for the case of $K = V$ (see Karger and Tai, 1997), to the best of our knowledge, there is no *efficient* way to generally approximate $u_G(P)$ with arbitrary precision ε in general graphs.

Let G_P denote a random subgraph of G in which every edge $e \in E$ is removed at random with probability p_e. Moreover, for an arbitrary graph $\mathscr{G}$, let $\phi(\mathscr{G})$ be a binary function such that $\phi(\mathscr{G}) = 1$ if the terminal set K of $\mathscr{G}$ is disconnected, and $\phi(\mathscr{G}) = 0$ otherwise. The simplest Monte Carlo estimator is as follows:

$$\hat{u}_G(P) = \frac{1}{N}\sum_{i=1}^{N} Y_i, \tag{1}$$

with $Y_i \sim \phi(G_P)$. Despite being straightforward, using the well known Chebyshev bound, one can prove that a sample size $N \in O(u_G(P)^{-1}\varepsilon^{-2}\log 1/\delta)$ is required to guarantee that $\hat{u}_G(P)$ is in the range $(1 \pm \varepsilon)u_G(P)$ with at least confidence of $1 - \delta$ (Dagum et al., 1995). This is problematic, as for highly reliable systems $u_G(P) \to 0$ and $N \to \infty$.

To overcome limitations of Eq. (1), researchers have leveraged rare-event simulation techniques. Among the most competitive approaches stand: the Multilevel Splitting method by Botev et al. (2012) and Zuev et al. (2015),[2] the Importance Sampling application of Cancela and El Khadiri (1995), the Permutation Monte Carlo-based method by Gertsbakh and Shpungin (2010) and its Splitting Sequential Monte Carlo extension (Vaisman et al., 2016). However, a recurring limitation for these techniques is that they appeal to the central limit theorem and assume non-trivially justified error distributions to draw confidence intervals. These assumptions do not hold in general, as some techniques return biased estimates, rely on Markov Chain Monte Carlo without specified mixing times, disregard finite sample size effects, or even suffer from numerical instability. Thus, upon their application, the user can neither guarantee, nor reliably quantify, the associated relative error and confidence of their approximations.

The absence of *principled* approximations in network reliability motivated the development of RelNet (Duenas-Osorio et al., 2017; Paredes et al., 2019)—a counting-based approach that is *guaranteed* to deliver estimates in the range $(1 \pm \varepsilon)u_G(P)$ with at least confidence $1 - \delta$, where $\varepsilon, \delta \in (0, 1)$ are user specified parameters. Despite success in power transmission network applications (Duenas-Osorio et al., 2017), such a methodology still independent edge failures. Also, empirical evaluations showed that small edge failure probabilities can increase runtime significantly.

Due to the current limitations, in this paper we lift RelNet to the weighted setting using importance sampling-based techniques. We show that the new approach can handle small edge failure probabilities while making no assumptions of independence.

The rest of the paper is as follows. Section 2 gives a refresher on RelNet (Paredes et al., 2019), highlighting its limitations on handling small edge failure probabilities as well as dependent edge failures. Section 3 introduces our importance sampling-based extensions to RelNet for supporting general weight functions. Section 4 presents numerical experiments employing benchmarks borrowed from the literature. We show that our developments render RelNet competitive in the rare event setting. Finally, Section 5 gives a summary of our findings and outlines future research directions.

2. Counting-Based Network Reliability

In this section we summarize background concepts in propositional logic. Then, we introduce the satisfiability (SAT) encoding of the K-terminal net-

[1] For true value of quantity a and approximate/inferred value $\bar{a}$, the relative error is $\varepsilon = |\bar{a}/a - 1|$.

[2] Subset simulation is an instance of the Markov Chain Monte Carlo-based *splitting* technique (Glasserman and Heidelberger, 1999)

work reliability problem. Later we discuss limitations handling small edge failure probabilities and dependent edge failures.

2.1. Background on Propositional Logic

A Boolean formula f over propositional variables $X = (x_1, \ldots, x_n)$, $X \in \{0,1\}^n$, is said to be in *conjunctive normal form* (CNF) when written as a conjunction of clauses $f = C_1 \wedge \cdots \wedge C_m$, where every clause C_i is a disjunction of literals,[3] e.g. $C_i = x_1 \vee \neg x_2 \vee x_3$. The *satisfiability* problem, or SAT, asks whether a CNF formula f is satisfiable or evaluate to true—exists $X \in \{0,1\}^n$ such that $f(X) = 1$. We define the set of satisfying assignments of f as $R_f = \{X \in \{0,1\}^n | f(X) = 1\}$. Moreover, the *model counting* problem, or #SAT, asks to compute $|R_f|$, i.e. the number of satisfying assignments.

We are interested in Σ_1^1 formulas, since they can handle global constraints, as is the case of network reliability. We say F is a Σ_1^1 formula if it is written in the form $F = \exists S[f(X,S]$, where f is a CNF formula over propositional variables $S = (s_1, \ldots, s_{n'})$, $S \in \{0,1\}^{n'}$, and $X \in \{0,1\}^n$ as before. Similarly, we define the set of satisfying assignments of F as:

$$R_F = \{X | \exists S \in \{0,1\}^{n'} \text{ such that } f(X,S) = 1\}. \quad (2)$$

The projected model counting problem, or #∃SAT, consists of computing $|R_F|$ given a Σ_1^1 formula F.

Towards model counting-based reliability estimation, we are interested in establishing a one-to-one relationship between the failed (not failed) states of a network and the truth (false) assignments of a Boolean formula. Moreover, we can represent every subgraph $G(X)$ of G via edge variables $X = (x_e)_{e \in E}$, $X \in \{0,1\}^{|E|}$, such that edge $e \in E$ is present in $G(X)$ if and only if $x_e = 1$. Our goal will be to find a formula[4] F over variables X and S such that $\phi(G(X)) \iff (\exists S) f(X,S)$, where ϕ is a binary function that outputs 1 if and only if the terminal set is disconnected. Then, for input graph $G(V,K,E)$ and edge failure probabilities $P = (1/2)_{e \in P}$, one can show that $u_G(P) = |R_F|/2^{|E|}$ (see Paredes et al., 2019). The next subsections show how to construct such a formula F and handle edge failure probabilities different than 1/2.

[3]A literal is a Boolean variable or its negation.

[4]In practice, we use a Σ_1^1 formula $F = (\exists S) f(X,S)$ due to auxiliary variables $S = (s_1, \ldots, s_{n'})$.

2.2. SAT-Encoding

Our formulation takes the form of a Σ_1^1 formula over propositional variables X and S, where variables in X are associated to the "real problem" and S contains auxiliary variables. In particular, for input graph $G(V,K,E)$, we introduce propositional variables $x_e \in X$ for every unreliable edge $e \in E$, and propositional variables $s_v \in S$ for every vertex $v \in V$. Then, the following Σ_1^1 formula is *equisatisfiable* with respect to the system failure indicator function ϕ:

$$F_K(X) = \exists S[f(X,S)] = \exists S\left[C_K \wedge C_{\neg K} \wedge C_E\right]. \quad (3)$$

$$C_K = \bigvee_{v \in K} s_v, \quad (4) \qquad C_{\neg K} = \bigvee_{v \in K} \neg s_v, \quad (5)$$

$$C_E = \bigwedge_{(a,b)=e \in E} (s_a \wedge x_e) \implies s_b, \quad (6)$$

An effective way to interpret $f(X,S)$ is to consider the subgraph $G(X) \subseteq G$ and picture every vertex $v \in V$ as "marked" if $s_v = 1$, and unmarked otherwise. Then, the clauses of Eq. (6) are satisfied whenever variable assignments of S are set so that neighbors of marked vertices are marked as well. Moreover, Eq. (4) [resp. Eq. (5)] admits variable assignments of S such that at least one terminal vertex is marked (resp. unmarked). Intuitively, if $G(X)$ is a subgraph of G such that $\phi(G(X)) = 0$,[5] then $f(X,S)$ is unsatisfiable. In other words, for all $X \in \{0,1\}^{|E|}$ such that $\phi(G(X)) = 0$, we cannot satisfy all clauses of f simultaneously. For example, C_K requires us to mark at least one vertex in K, yet, regardless of what subset of K we choose to "mark", C_E then forces us to "propagate" this marking across every vertex in K because the terminal set in $G(X)$ is connected—making it impossible to satisfy $C_{\neg K}$.

For edge failure probabilities of 1/2, it was shown that $u_G(P) = R_{F_K}/2^{|E|}$ (Paredes et al., 2019). Thus, upon construction of F_K, we approximate

[5]By definition, the terminal set of $G(X)$ is connected when $\phi(G(X)) = 0$.

$u_G(P)$ with PAC guarantees leveraging state-of-the-art *model counters*—typically used in software and hardware verification, as well as automation and data driven applications in robotics, air traffic, Bayesian models, etc. (Soos and Meel, 2019). The next subsection gives a refresher on how to handle arbitrary edge failure probabilities and practical challenges.

2.3. *The Weighted-to-Unweighted Reduction*

An important assumption in the current RelNet framework is that edges fail *independently* with probability 1/2. Next, we show how arbitrary failure probabilities can be reduced to such case. The key observation is that every edge $e = (a,b)$ in E, with failure probability $p_e \neq 1/2$, can be replaced with a reliability preserving series-parallel graph $G_e(V_e, K_e = \{a,b\}, E_e)$ in which edges fail with probability 1/2. Here, reliability preserving means that $p_e = u_{G_e}(P_e)$. The construction of V_e and E_e is described elsewhere (Paredes et al., 2019). It was shown that the size of G_e scales linearly in the size of the binary representation of the edge reliability $1 - p_e$. For example, the binary representation of edge reliability 0.9375 is 0.1111, and G_e takes the form of a parallel graph with 2 vertices and 4 edges that fail with probability 1/2.

Small probabilities and dependent edge failures can render RelNet impractical. These limitations inspire us to lift RelNet to support general weight functions. The next section introduces weighted model counting as well as RelNet developments to handle this extended formalism.

3. WEIGHTED RELNET

For a CNF formula f over variables $X = \{0,1\}^n$ and set of satisfying assignments R_F, a generalization of counting is that of considering a weight function $W : \{0,1\}^n \to \mathbb{R}$. Then, the goal in *weighted* model counting is to compute:

$$\mathscr{W}(R_f) = \sum_{X \in \{0,1\}^n} W(X) \cdot f(X) = \sum_{X \in R_f} W(X), \quad (7)$$

where R_f is the set of satisfying assignments as defined earlier. In the specific case of the K-terminal network reliability problem, the weight function becomes the joint probability density function (PDF) of $X \in \{0,1\}^{|E|}$, i.e.:

$$W(X) = \Pr(X) = \prod_{e \in E} (1 - p_e)^{x_e} \cdot p_e^{x_e}. \quad (8)$$

However, throughout this paper we assume that $W(X)$ is black box, admitting joint PDFs with general correlation structure. In addition to weighted model counting, we are interested in the case of *projected* model counting for handling formulas such as RelNet's encoding of Eq. (3). The generalization can be integrated into Eq. (7) by substituting R_f for R_F as defined in Section 2.1. Despite weighted and projected model counting being each active areas of research, there is a limited number of tools that can handle both tasks simultaneously in practice. In particular, a general hashing-based framework called ApproxMC can be readily used in this complex setting. The next subsections introduce the ApproxMC framework and importance sampling strategies used in weighted RelNet.

3.1. *Hashing Framework for Sampling*

Significant advances in modern SAT solvers have enabled their use as *oracles* in model counting algorithms. In fact, for formulas with thousands of variables, a solver[6] can aid enumerating all members of R_f (or R_F). Nevertheless, when R_f is of exponential size, the naive approach is intractable. Previous theoretical work hypothesized the use of *hash functions* as an *efficient* way to approximate counting problems. Thus, building upon previous work and leveraging modern SAT solvers, ApproxMC was introduced as a general hashing-based approximation framework. Next we discuss some of the core ideas of this method.

For variables $X = (x_1, \ldots, x_n)$, let us introduce the next family of functions:

$$\mathscr{H} = \left\{ (X \cdot b)_{\%2} \middle| b \in \{0,1\}^n \right\} \quad (9)$$

where "·" denotes the dot product and "%2" the modulo 2 operation. For $h \in \mathscr{H}$ and $y \in \{0,1\}$, the expression $h(X) = y$ is typically referred to as an *XOR* constraint. The key insight is that choosing $h(X) = y$ at random and considering $F'(X) = F(X) \wedge \big(h(X) = y\big)$ has the effect of *roughly* halving

[6]For example: CryptoMiniSAT.

the number of satisfying assignments, i.e $|R_{F'}| \approx \frac{1}{2}|R_F|$. Next we expand this idea further. Let us define the next family of hash functions:

$$\mathscr{H}_m = \{H = (h_1, \ldots, h_m) | h_i \in \mathscr{H}\}. \qquad (10)$$

Note that for $H_m \in \mathscr{H}_m$, $H_m(X)$ outputs an m-bit vector, effectively partitioning the domain into 2^m *cells*, i.e. $H_m : X \in \{0,1\}^n \to y \in \{0,1\}^m$. Moreover, by choosing $y \in \{0,1\}^m$ at random, we can randomly select a cell via the set of XOR constraints $H_m(X) = y$. In addition, for $F' = F \wedge \big(H_m(X) = y\big)$, Meel (2017) showed with probabilistic guarantees that $|R_F| \approx |R_{F'}| \times 2^m$. Thus, in practice, one can choose $m \in \{0, \ldots, n\}$ large enough such that $|R_{F'}|$ is small—and hence the number of solver calls—but not too small to avoid large deviations from $|R_F|$ when $|R_{F'}|$ is scaled by 2^m. ApproxMC is an state-of-the-art algorithm that formalizes these ideas and can handle formulas with several thousands of variables (Meel, 2017).

In Section 4, we show empirically that ApproxMC can handle challenging instances of the K-terminal network reliability problem after encoding them into Σ_1^1 formulas using RelNet. However, general correlation structures are not supported and, when edge failure probabilities are very small, the performance of ApproxMC deteriorates. The next subsection introduces our suggested strategies to overcome these issues.

3.2. *Hashing-based Importance Sampling*

In general, let the domain $\Omega = \{0,1\}^n$ be partitioned into disjoint subsets, or *cells*, $\Omega_0, \ldots, \Omega_{2^m-1} \subseteq \Omega$, with $0 \leq m \leq n$. Letting $\mathbb{1}_\Omega(X)$ denote the indicator function of set membership, i.e. $\mathbb{1}_\Omega(X)$ outputs 1 if $X \in \Omega$ and 0 otherwise, we can write $\mathscr{W}(R_F)$ of Eq. (7) in the alternate form:

$$\mathscr{W}(R_F) = \sum_{c=0}^{2^m-1} \left(\sum_{X \in R_F} W(X) \cdot \mathbb{1}_{\Omega_c}(X) \right). \qquad (11)$$

We can construct an estimator of $\mathscr{W}(R_F)$ choosing subset Ω_c with probability μ_c and computing:

$$\widehat{\mathscr{W}(R_F)} = \frac{1}{\mu_c} \times \sum_{X \in R_F} W(X) \cdot \mathbb{1}_{\Omega_c}(X). \qquad (12)$$

It is easy to show that Eq. (12) unbiased, i.e. $\mathbb{E}[\widehat{\mathscr{W}(R_F)}] = \mathscr{W}(R_F)$. Also, for y_c the m-bit encoding of integer c, with $0 \leq c \leq 2^m - 1$, note that every $H_m \in \mathscr{H}_m$ partitions Ω into 2^m subsets. Formally, $H_m : X \in \{0,1\}^m \to y_c \in \{0,1\}^m$. Thus, $\mathbb{1}_{\Omega_c}(X) = \mathbb{1}\big(H_m(X) = y_c\big)$. Next, we show two straightforward applications of the hashing framework for approximating $\mathscr{W}(R_F)$.

3.2.1. *Uniform cell sampling*

Sampling an m-bit vector, in which "0-1" bits have equal chance, will choose a cell $y_c \in \{0,1\}^m$ uniformly and at random with probability $\mu_c = 1/2^m$. Setting $F_c = F \wedge \big(H_m(X) = y_c\big)$, the estimator of Eq. (12) becomes:

$$\widehat{\mathscr{W}(R_F)} = 2^m \times \sum_{X \in R_{F_c}} W(X). \qquad (13)$$

We term this estimator UXOR. A similar version of this estimator has been used in the past (Chakraborty et al., 2014). However, for $w_{\min} = \min_{X \in R_F} W(X)$ and $w_{\max} = \max_{X \in R_F} W(X)$, the number of SAT solver invocations is bounded by $\rho = w_{\max}/w_{\min}$. This is problematic in network reliability, as for n_C denoting the size of the minimum K-cut, and assuming that all edges fail with probability $p < 0.5$, we have that $\rho = [(1-p)/p]^{|E|-n_C}$, i.e. $\rho \to \infty$ as $p \to 0$. This motivated us to sample cells non-uniformly.

3.2.2. *Biased cell sampling*

Assume that we are able to sample $y_c \in \{0,1\}^m$, or its associated subset Ω_c, with probability $\Pr(X \in \Omega_c)$. Then, our estimator becomes:

$$\widehat{\mathscr{W}(R_F)} = \frac{1}{\Pr(X \in \Omega_c)} \times \sum_{X \in R_{F_c}} W(X), \qquad (14)$$

with $F_c = F \wedge \big(H_m(X) = y_c\big)$. We term the estimator BXOR. To use Eq. (14) we need to address: (i) sampling of y_c, and (ii) computing $\Pr(X \in \Omega_c)$. For the purpose of (i) and (ii) we assume $\Pr(X)$ has the form of Eq. (8), but $W(X)$ can be of general form. For (i) we sample X' with $\Pr(X')$ as in the crude Monte Carlo approach, then we select cell $y_c = H_n(X')$, since $X' \in \Omega_c$ with $\Pr(X' \in \Omega_c)$. For (ii) we need to do more work.

First, let $H_m(X|x_1,\dots,x_{m'})$, $0 \le m' \le n$, denote H_m when each variable $x_1,\dots,x_{m'}$ has been replaced with a value in $\{0,1\}$. We compute $\Pr(X \in \Omega_c)$ by branching on variables $x_1,\dots,x_n$ one at a time and keeping track of the systems of XOR constraints. Moreover, let us write the system of equations $H_m(X|x_1,\dots,x_{m'}) = y_c$ more conveniently by working all constant terms towards the right hand side. For example, $(1+0+x_4+x_8)_{\%2} = 1$ becomes $(x_4+x_8)_{\%2} = 0$. We shall denote this form $H'_m(X|x_1,\dots,x_{m'}) = y'_c$. Note that, regardless of the assignment of variables $x_1,\dots,x_{m'}$, the term $H'_m(X|x_1,\dots,x_{m'})$ is fixed, while the right hand side term is $y'_c \in \{0,1\}^m$. Thus, at any moment, we need to keep track of at most 2^m terms. We use a data structure $p' : \{0,1\}^m \to [0,1]$ to collect their associated probabilities. Clearly, after we have branched over all variables, the left hand side is $H'_m(X|x_1,\dots,x_{n'}) = \{0\}^m$. Thus, we return $\Pr(X \in \Omega_c) = p'(\{0\}^m)$ and plug it into Eq. (14).

Since the computation of $\Pr(X \in \Omega_c)$ scales as 2^m, it can become a bottleneck for problems requiring a large number of XOR constraints m. Next we show an additional form of importance sampling that can alleviate this issue.

3.3. *Distributional Importance Sampling*

A standing issue in the two precedent approaches is that the amount of cells $y_c \in \{0,1\}^m$ is exponential in the number of XOR constrains $m \in \{0,\dots,n\}$. Despite the number of satisfying assignments being nearly uniform across cells, the *weighted* count is highly variable. This is a challenge in practice, as sample sizes need to be intractably large. Thus, to constraint m, we use a sampling approach that reduces the number of variables of the problem. Distributional importance sampling (Davies and Bacchus, 2008), partitions variables $X = (x_1,\dots,x_n)$ into $X_p = (x_1,\dots,x_p)$ and $X_{\bar{p}} = (x_{p+1},\dots,x_n)$, then it samples variables X_p using a proposal distribution and lets the other variables unassigned. The resulting formula is feed to a weighted model counter, and the resulting output is properly weighted by an importance sampling factor. We rewrite Eq. (7) as follows:

$$\mathscr{W}(R_F) = \sum_{X_p \in \{0,1\}^p} \sum_{X_{\bar{p}} \in \{0,1\}^{(n-p)}} W(X) \cdot F(X). \quad (15)$$

Table 1: RelNet *results for the dodecahedron topology with homogeneous edge failure probabilities setting* $(\varepsilon,\delta) = (0.8, 0.2)$.

q	$\hat{u}_G(P)$	ε_o	Time (s)
10^{-1}	2.502×10^{-03}	0.003	426
10^{-2}	2.056×10^{-06}	0.008	791
10^{-3}	2.095×10^{-09}	0.046	1438
10^{-4}	2.018×10^{-12}	0.009	2673
10^{-5}	2.109×10^{-15}	0.055	2903

Then, the distributional importance sampling (DIS) estimator consists of sampling X_p with probability $\mu(X_p)$ and computing:

$$\widehat{\mathscr{W}(R_F)} = \frac{1}{\mu(X_p)} \sum_{X_{\bar{p}} \in \{0,1\}^{(n-p)}} W(X) \cdot F(X). \quad (16)$$

This approach has been used for weighted model counting, but leveraging the ApproxMC framework we apply it in the most general case of interest, that of weighted and projected model counting. Also, we can *compound* unbiased estimators, e.g. UXOR or BXOR, with DIS at the price of increasing the variance (Karger, 2017, Lemma II.3). Next, we test these importance sampling strategies using popular benchmarks in network reliability.

4. Computational Experiments

In our experiments we use ApproxMC (Soos and Meel, 2019), a flexible tool that can integrate ideas of Section 3 to return weighted and projected model count approximations. An important parameter regarding the speed and accuracy of ApproxMC is the *threshold*—the number of solutions considered in the final approximation. Typically, the threshold value depends on the user specified error guarantees $\varepsilon, \delta \in (0,1)$, as in the case of unweighted RelNet. However, for Weighted-RelNet via importance sampling estimators UXOR+DIS and BXOR+DIS, we set the *threshold* high enough so that we obtain a small *coefficient of variation* (*CoV*), defined as $\bar{\sigma}/\bar{\mu}$, with $\bar{\mu}$ the sample mean and $\bar{\sigma}$ the sample standard deviation. Next we describe specific settings, benchmarks, and results using RelNet and Weighted-RelNet.

*Table 2: Weighted-*RelNet *setting threshold to 4096 and repetitions to 10,000.*

q	$\hat{u}_G(P)$	ε_o	CoV	Time (s)
		UXOR+DIS		
10^{-1}	2.272×10^{-03}	0.089	0.065	1322
10^{-2}	3.474×10^{-06}	0.702	0.370	1326
10^{-3}	2.731×10^{-09}	0.363	0.569	1349
10^{-4}	3.409×10^{-12}	0.704	0.471	1416
10^{-5}	1.428×10^{-15}	0.286	0.217	1401
		BXOR+DIS		
10^{-1}	2.234×10^{-03}	0.104	0.091	1290
10^{-2}	1.968×10^{-06}	0.035	0.153	1318
10^{-3}	3.422×10^{-09}	0.708	0.461	1328
10^{-4}	1.608×10^{-12}	0.196	0.195	1386
10^{-5}	1.450×10^{-15}	0.275	0.206	1337

4.1. *Dodecahedron topology*

The dodecahedron graph shown in Figure 1 has been widely used among stochastic simulation researchers (e.g. Cancela and El Khadiri, 1995; Botev et al., 2012; Vaisman et al., 2016). We consider homogeneous edge failure probabilities q, and let $q = 10^{-1}, \ldots, 10^{-5}$. The configuration of RelNet is as follows. For the unweighted case we set $(\varepsilon, \delta) = (0.8, 0.2)$. For the weighted case, we use $p = 16$ in DIS,[7] fixed *threshold* $= 4096$ and found relatively good values of CoV with $N = 10,000$ samples.

As evidenced in Table 1, unweighted RelNet delivers approximations that meet the specified input guarantees. In fact, the observed relative error, denoted ε_o, far outperforms the input $\varepsilon = 0.8$. However, as edge failure probabilities become smaller, the runtime significantly increases due to the weighted-to-unweighted reduction.

Table 2 summarizes results for weighted RelNet. The runtime does not increases significantly because the counting routines do not depend on the edge failure probabilities. Also, we have fixed the number of samples. While our empirical results favor BXOR over UXOR, overall results are similar. Both verify ε_o values smaller than 0.8, but the CoV is still relatively large when compared to those typically sought in practice—less than 0.05.

[7]In our experiments, $p < |E| - 14$ renders the computation of $\Pr(X \in \Omega_c)$ in BXOR too expensive.

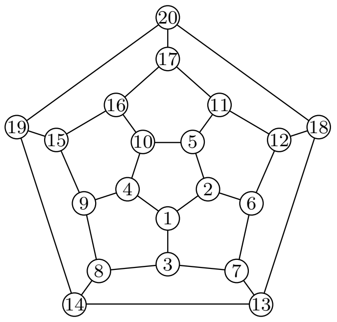

Figure 1: The dodecahedron graph. $K = \{1,3\}$.

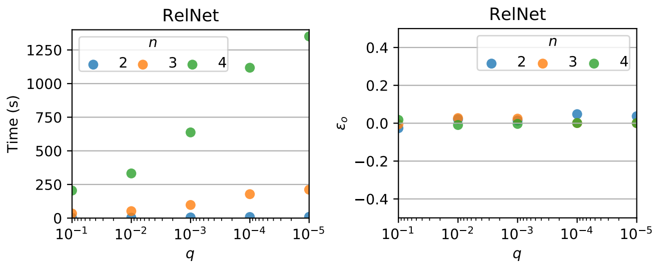

Figure 2: RelNet *results in the grid topology. Left: runtime in seconds. Right: signed relative error.*

4.2. *Grid Networks*

We also consider the $n \times n$ square grid and let the terminal set contain two vertices at opposing corners. This topology can be grown arbitrarily large via n. Similarly, we let all edges fail with small probabilities of $q = 10^{-1}, \ldots, 10^{-5}$, and consider values of $n = 2, 3, 4$. The configuration of the RelNet variants is as before, except for p in DIS, which is set to $p = \max(0, |E| - 14)$.

Figure 2 shows results for RelNet. In particular, small edge failure probabilities have a significant impact in runtime due to the weighted-to-unweighted reduction. Also, RelNet far exceeded the expectations in terms of the observed signed relative error, defined as $\bar{u}/u - 1$.

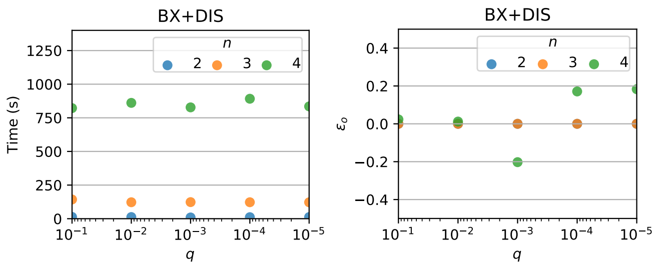

Figure 3: BXOR+DIS *results in the grid topology. Left: runtime in seconds. Right: signed relative error.*

Figure 3 shows only results for BXOR+DIS, but we obtained similar results for UXOR+DIS with slightly worse results in terms of the observed relative error. As expected, small edge failure probabilities did not impact runtime. Also, while somewhat larger than RelNet, the observed signed relative error values remained far below 0.8.

5. Conclusions

This paper introduced Weighted-RelNet, a counting-based approach to evaluate critical infrastructure reliability. We showed empirically that our framework, which rests upon importance sampling principles, can handle small failure probabilities. Its black box formulation of the weight function makes it unique for handling systems with general correlation structure. A rigorous theoretical analysis of the current techniques should be pursued for endowing Weighted-RelNet with provable guarantees of approximation—currently available in the unweighted approach.

Future work can focus on developing a hashing-based importance sampling approach that choses cells with probability proportional to cells' satisfying assignment weights, instead of cells' assignment likelihood. This would result in a significant reduction of variance. Moreover, computational explorations using dependent failures should be carried out to evaluate the competitiveness of weighted RelNet in this setting, which is known to challenge efficient recursive techniques.

6. Acknowledgments

The authors gratefully acknowledge the support by the U.S. Department of Defense (Grant W911NF-13-1-0340) and the U.S. National Science Foundation (Grants CMMI-1436845 and CMMI-1541033).

7. REFERENCES

Ball, M. O., Colbourn, C. J., and Provan, J. S. (1995). "Chapter 11 Network reliability." *Handbooks in Operations Research and Management Science*, Vol. 7, 673–762.

Botev, Z. I., L'Ecuyer, P., Rubino, G., Simard, R., and Tuffin, B. (2012). "Static Network Reliability Estimation via Generalized Splitting." *INFORMS Journal on Computing*, 25(1), 56–71.

Cancela, H. and El Khadiri, M. (1995). "A recursive variance-reduction algorithm for estimating communication-network reliability." *IEEE Transactions on Reliability*, 44(4), 595–602.

Chakraborty, S., Fremont, D. J., Meel, K. S., Seshia, S. A., and Vardi, M. Y. (2014). "Distribution-Aware Sampling and Weighted Model Counting for SAT." *AAAI*, 1722–1730 (apr).

Dagum, P., Karp, R., Luby, M., and Ross, S. (1995). "An optimal algorithm for Monte Carlo estimation." *Proceedings of IEEE 36th Annual Foundations of Computer Science*, Vol. 29, IEEE Comput. Soc. Press, 142–149 (jan).

Davies, J. and Bacchus, F. (2008). "Distributional importance sampling for approximate weighted model counting." *Workshop on Counting Problems in CSP and SAT, and other neighbouring problems.*

Duenas-Osorio, L., Meel, K., Paredes, R., and Vardi, M. (2017). "Counting-based reliability estimation for power-transmission grids." *Aaai*, San Francisco, 4488–4494.

Gertsbakh, I. B. and Shpungin, Y. (2010). *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo.* CRC Press.

Glasserman, P. and Heidelberger, P. (1999). "Multilevel Splitting for Estimating Rare Event Probabilities." *Operations Research*, 47(4), 585–600.

Karger, D. and Tai, R. (1997). "Implementing a fully polynomial time approximation scheme for all terminal network reliability." *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, 334–343.

Karger, D. R. (2017). "Faster (and Still Pretty Simple) Unbiased Estimators for Network (Un)reliability." *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, 755–766 (oct).

Meel, K. S. (2017). "Constrained Counting and Sampling: Bridging the Gap Between Theory and Practice." Ph.D. thesis, Rice University, Rice University.

Paredes, R., Duenas-Osorio, L., Meel, K. S., and Vardi, M. Y. (2019). "Principled Network Reliability Approximation: A Counting-Based Approach." *Reliability Engineering & System Safety (submitted).*

Soos, M. and Meel, K. S. (2019). "Bird: Engineering an efficient cnf-xor sat solver and its applications to approximate model counting." *Proceedings of AAAI Conference on Artificial Intelligence (AAAI).*

Vaisman, R., Kroese, D. P., and Gertsbakh, I. B. (2016). "Splitting sequential Monte Carlo for efficient unreliability estimation of highly reliable networks." *Structural Safety*, 63, 1–10.

Zuev, K. M., Wu, S., and Beck, J. L. (2015). "General network reliability problem and its efficient solution by Subset Simulation." *Probabilistic Engineering Mechanics*, 40, 25–35.