Reliability Engineering and System Safety 191 (2019) 106472

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Principled network reliability approximation: A counting-based approach

R. Paredes*[,a], L. Dueñas-Osorio[a], K.S. Meel[b,c], M.Y. Vardi[c]

[a] Department of Civil and Environmental Engineering, Rice University, Houston, USA
[b] Computer Science Department, National University of Singapore, Singapore
[c] Department of Computer Science, Rice University, Houston, USA

ARTICLE INFO

Keywords:
Network reliability
FPRAS
PAC
Relative variance
Uncertainty
Model counting
Satisfiability

ABSTRACT

As engineered systems expand, become more interdependent, and operate in real-time, reliability assessment is key to inform investment and decision making. However, network reliability problems are known to be #P-complete, a computational complexity class believed to be intractable, and thus motivate the quest for approximations. Based on their theoretical foundations, reliability evaluation methods can be grouped as: (i) *exact or bounds*, (ii) *guarantee-less sampling*, and (iii) *probably approximately correct* (PAC). Group (i) is well regarded due to its useful byproducts, but it does not scale in practice. Group (ii) scales well and verifies desirable properties, such as the *bounded relative error*, but it lacks error guarantees. Group (iii) is of great interest when precision and scalability are required. We introduce $\mathcal{K}$-RelNet, an extended counting-based method that delivers PAC guarantees for the $\mathcal{K}$-terminal reliability problem. We also put our developments in context relative to classical and emerging techniques to facilitate dissemination. Then, we test in a fair way the performance of competitive methods using various benchmark systems. We note the range of application of algorithms and suggest a foundation for future computational reliability and resilience engineering, given the need for principled uncertainty quantification across complex networked systems.

1. Introduction

Modern societies rely on physical and technological networks such as transportation, power, water, and telecommunication systems. Quantifying their reliability is imperative in design, operation, and resilience enhancement. Typically, networks are modeled using a graph where vertices and edges represent unreliable components. Network reliability problems ask: what is the probability that a complex system with unreliable components will work as intended under prescribed functionality conditions?

In this paper, we focus on the $\mathcal{K}$-terminal reliability problem [1]. In particular, we consider an undirected graph $G = (V, E, \mathcal{K})$, where V is the set of vertices, $E \subseteq V \times V$ is the set of edges, and $\mathcal{K} \subseteq V$ is the set of terminals. We let $G(P)$ be a stochastic graph, where every edge $e \in E$ vanishes from G with respective probabilities $P = (p_e)_{e \in E}$. We assume a binary system, and say $G(P)$ is *unsafe* if a subset of vertices in $\mathcal{K}$ becomes disconnected, and *safe* otherwise. Thus, given an instance (G, P) of the $\mathcal{K}$-terminal reliability problem, we are interested in computing the unreliability of $G(P)$, denoted $u_G(P)$, and defined as the probability that $G(P)$ is unsafe.

If $|\Theta|$ is the cardinality of set Θ, then $n = |V|$ and $m = |E|$ are the number of vertices and edges, respectively. Also, when $|\mathcal{K}| = n$ and $|\mathcal{K}| = 2$, the $\mathcal{K}$-terminal reliability problem reduces to the all-terminal and two-terminal reliability problems, respectively. These are well-known and proven to be #P-complete problems [1,2]. The more general $\mathcal{K}$-terminal reliability problem is #P-hard, so ongoing efforts to compute $u_G(P)$ focus on practical bounds and approximations.

Exact and bounding methods are limited to networks of small size, or with bounded graph properties such as treewidth and diameter [3,4]. Thus, for large G of general structure, researchers and practitioners lean on simulation-based estimates with acceptable Monte Carlo error [5]. However, in the absence of an error prescription, simulation applications can use unjustified sample sizes and lack a priori rigor on the quality of the estimates, thus becoming *guarantee-less* methods.

A formal approach to guarantee quality in Monte Carlo applications relies on the so-called (ϵ, δ) *approximations*, where ϵ and δ are user specified parameters regarding the relative error and confidence, respectively. As an illustration, for Y as a random variable (RV), say we are interested in computing its expected value $E[Y] = \mu_Y$. Then, after *we specify* parameters $\epsilon, \delta \in (0, 1)$, an (ϵ, δ) approximation returns estimate $\bar{\mu}_Y$ such that $\Pr(|\bar{\mu}_Y/\mu_Y - 1| \geq \epsilon) \leq \delta$. In other words, an (ϵ, δ) *approximation* returns an estimate with relative error below ϵ with at least confidence $1 - \delta$. We term *Probably Approximately Correct* (PAC) the family of methods whose algorithmic procedures deliver estimates

* Corresponding author.
E-mail address: roger.paredes@rice.edu (R. Paredes).

https://doi.org/10.1016/j.ress.2019.04.025
Received 4 June 2018; Received in revised form 16 March 2019; Accepted 20 April 2019
Available online 25 April 2019

with (ϵ, δ) guarantees.[1]

Having a formal notion of error, we can rigorously address a key issue in Monte Carlo applications: the *sample size*, herein denoted N. Using standard probability arguments, and positive finite μ_Y as the only assumption,[2] we derive: $N = O(\sigma_Y^2/\mu_Y^2 \epsilon^{-2} \log 1/\delta)$ (see appendix, Theorem 6), exposing Monte Carlo's weakness when required to guarantee results. To make it self-evident, let us model our binary-system as Y, a Bernoulli RV, such that $\mu_Y = u_G(P)$. Then, note that the substitution of μ_Y and σ_Y^2 in N leads to $N \propto 1/u_G(P)$, which can be prohibitively large as engineered systems are meant to be *highly-reliable* by design.

The sample size issue is a well researched subject of *rare-event* simulation, and we refer readers to Chapter 2 [7] and Chapter 1 [8] for more background. Attempts to make simulation more affordable include: the Multilevel Splitting method [9,10], the recursion-based Importance Sampling method [11], the Permutation Monte Carlo-based method [12] and its Splitting Sequential Monte Carlo extension [13], among others. Some of these techniques verify desired properties, such as the Bounded Relative Variance (BRV) or $\sigma_Y^2/\mu_Y^2 = O(1)$, and the Vanishing Relative Variance (VRV) or $\sigma_Y^2/\mu_Y^2 = o(1)$, where Y denotes the Monte Carlo estimate returned by a sampling technique.[3] Despite being effective in the rare-event setting, these methods often appeal to the central limit theorem and do not assure quality of error or performance, thus remaining guarantee-less to users.

Naturally, a method that overcomes the rare-event issue while delivering rigorous error guarantees would be of great use in reliability applications. In other words, system reliability is calling for *efficient* PAC methods for a rigorous treatment of uncertainties. Theoretically speaking, an efficient method runs in polynomial time as a function of the size of (G, P), $1/\epsilon$, and $\log(1/\delta)$. In the computer science literature, such a routine is called a *fully polynomial randomized approximation scheme* (FPRAS) for network unreliability. Clearly, efficient in theory does not imply efficient in practice, e.g., the order of the polynomial function bounding the worst-time complexity can be arbitrarily large. Thus, it is imperative to complement theoretically sound developments with computer evaluations. To the best of our knowledge, there is no known FPRAS for the $\mathcal{K}$-terminal reliability problem. However, there is a precedent, where Karger gave the first FPRAS for the all-terminal reliability case [14].

To tackle computational and precision issues, this paper develops $\mathcal{K}$-RelNet, a counting-based PAC method for network unreliability that inherits properties of state-of-the-art approximate model counters in the field of computational logic [15]. Our approach delivers rigorous (ϵ, δ) guarantees and is *efficient* when given access to an NP-oracle: a black-box that solves nondeterministic polynomial time decision problems. The use of NP-oracles for randomized approximations, first proposed by Stockmeyer [17], is increasingly within reach as in practice we can leverage efficient solvers for Boolean satisfiability (SAT) and treat them as oracles, given they are under active development. Given the variety of methods to compute $u_G(P)$, we showcase our developments against alternative approaches. In the process, we highlight methodological connections missed in the engineering reliability literature, key theoretical properties of our method, and unveil practical performance through fair computational experiments by using existing and our own benchmarks.

The rest of the manuscript is structured as follows: Section 2 gives background on network reliability evaluation and its (ϵ, δ) approximation, as well as the necessary background on Boolean logic before introducing our new counting-based approach: $\mathcal{K}$-RelNet, an efficient PAC method for the $\mathcal{K}$-terminal reliability problem. Section 3 contextualizes our contribution relative to other techniques for network reliability evaluation. We highlight key properties for users and draw important connections in the literature. Section 4 presents the main results of our computational evaluation. Section 5 rounds up this study with conclusions and promising research directions.

2. Counting-based network reliability evaluation

We begin this section with relevant mathematical background and notation, then we introduce the new method, termed $\mathcal{K}$-RelNet. We do so through a fully worked out example for counting-based reliability estimation.

2.1. Principled network reliability approximation

Given instance (G, P) of the $\mathcal{K}$-terminal reliability problem, we represent a realization of the stochastic graph $G(P)$ as an m-bit vector $X = (x_e)_{e \in E}$, with $m = |E|$, such that $x_e = 0$ if edge $e \in E$ is in a failed state and $x_e = 1$ otherwise. Note that $\Pr(x_e = 0) = p_e$, and that the set of possible realizations is $\Omega = \{0, 1\}^m$. Furthermore, let Φ: $\Omega \mapsto \{0, 1\}$ be a function such that $\Phi(X) = 0$ if some subset of $\mathcal{K}$ becomes disconnected, i.e. X is *unsafe*, and $\Phi(X) = 1$ otherwise. Also, we define the *failure* and *safe* domains as $\Omega_f = \{X \in \Omega: \Phi(X) = 0\}$ and $\Omega_s = \{X \in \Omega: \Phi(X) = 1\}$, respectively. In practice, we can evaluate Φ efficiently using breadth-first-search.

Network reliability, denoted as $r_G(P)$, can be computed as follows:

$$r_G(P) = 1 - u_G(P) = \sum_{X \in \Omega} \Phi(X) \cdot \Pr(X), \tag{1}$$

$$\Pr(X) = \prod_{e_i \in E} p_{e_i}^{(1-x_i)} \cdot (1 - p_{e_i})^{x_i}, \tag{2}$$

where Eq. (2) assumes independent edge failures. Clearly, the number of terms $|\Omega| = 2^m$ of Eq. (1) grows exponentially, rendering the brute-force approach useless in practice, and motivating the development of network reliability evaluation methods that can be grouped into: exact or bounds, guarantee-less simulation, and probably approximately correct (PAC).

When exact methods fail to scale in reliability calculations, simulation is the preferred alternative. However, mainstream applications of simulation lack performance guarantees on error and computational cost. Typically, users embark on a trial and error process for choosing the sample size, trying to meet, if at all possible, a target empirical measure of variance such as the coefficient of variation. However, similar approaches have been shown to be unreliable [18], jeopardizing reliability applications at a time when uncertainty quantification is key, as systems are increasingly complex [19].

To secure a rigorous application of the Monte Carlo method, we use (ϵ, δ) approximation methods, which use no assumptions such as the central limit theorem, and that give guarantees of approximation in the non-asymptotic regime, i.e., they deliver *provably* sound approximations with a finite number of samples. Formally, for input parameters ϵ, $\delta \in (0, 1)$, we define a PAC method for network unreliability evaluation as one that outputs estimate $\hat{u}_G(P)$ such that:

$$\Pr\left(\frac{|\hat{u}_G(P) - u_G(P)|}{u_G(P)} \geq \epsilon \right) \leq \delta. \tag{3}$$

Recently, the authors introduced RelNet [20], a counting based framework for approximating the two-terminal reliability problem that issues (ϵ, δ) guarantees. In this paper, we introduce $\mathcal{K}$-RelNet, an extension that, to the best of our knowledge, is the first *efficient* PAC method for the general $\mathcal{K}$-terminal reliability problem.

Next, we survey important background in Boolean logic definitions before introducing $\mathcal{K}$-RelNet.

2.2. Boolean logic

A Boolean formula ψ: $X \in \{0, 1\}^n \rightarrow \{0, 1\}$ is in conjunctive normal

[1] We borrow the PAC terminology from the field of artificial intelligence [6].

[2] In this paper μ_Y will be a probability, such as network unreliability $u_G(P)$.

[3] Where the little-o notation $f(n) = o(g(n))$ stands for $f/g \rightarrow 0$, for $n > n_0$, $n, n_0 \in \mathbb{N}$.

form (CNF) when written as $\psi(X) = C_1 \wedge \cdots \wedge C_m$, with each clause C_i a disjunction of literals, e.g., $C_1 = x_1 \vee \neg x_2 \vee x_3$. We are interested in solving the #SAT ("Sharp SAT") problem, which counts the number of variable assignments satisfying a CNF formula. Formally, $\#\psi = |\{X \in \{0, 1\}^n | \psi(X) = 1\}|$. For example, consider the expression $x_1 \neq x_2$. Its CNF representation is $\psi(X) = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$, and the number of satisfying assignments of ψ is $\#\psi = 2$.

Furthermore, for Boolean vectors of variables $X = (x_1, \cdots, x_n)$ and $S = (s_1, \cdots, s_p)$, define a Σ_1^1 formula as one that is expressed in the form $F(X, S) = \exists S[\psi(X, S)]$, with ψ a CNF formula over variables X and S. Similarly, we are interested in its associated counting problem, called projected counting or "#∃SAT." Formally, $\#F = |\{X \in \{0, 1\}^n | \exists S \text{ such that } \psi(X, S) = 1\}|$. We use Σ_1^1 formulas because they let us introduce needed auxiliary variables (S) for global-level Boolean constraints, required in network reliability, but count strictly over the problem variables (X). As an example, consider the expression $[(x_1 \neq x_2) \text{ OR } (s_1 \neq x_2)]$. Its CNF representation is $\psi(X, S) = (x_1 \vee s_1 \vee x_2) \wedge (\neg x_1 \vee \neg s_1 \vee \neg x_2)$, and note the difference between the associated counts $\#\psi = 6$ and $\#F = 4$. The latter is smaller because the quantifier ∃ over variables S "projects" the count over variables X. To better grasp this projection, observe that $F(X, S) = \exists S[\psi(X, S)]$ is *equivalent* to which in our example simplifies to $(\neg x_1 \vee \neg x_2) \vee (x_1 \vee x_2) = 1$, i.e., for every assignment of variables $X \in \{0, 1\}^2$, there is $S \in \{0, 1\}$ such that $F(X, S) = 1$, and thus $\#F = 4$. The equivalent form is shown only for illustration purposes, as it is intractable to work with it explicitly due to its length growing exponentially in the number of variables in S. Instead, we feed $F(X, S) = \exists S[\psi(X, S)]$ to a state-of-the-art approximate model counter, such as ApproxMC3 [21].

Next, we introduce $F_{\mathcal{K}}$, a Σ_1^1 formula encoding the unsafe property of a graph G, and show that $\#F_{\mathcal{K}} = |\Omega_f|$. Recall Ω_f is the network failure domain $\Omega_f = \{X \in \Omega: \Phi(X) = 0\}$. Moreover, using a polynomial-time reduction to address arbitrary edge failure probabilities, we solve the $\mathcal{K}$-terminal reliability problem by computing $\#F_{\mathcal{K}}$. The problem of counting the number of satisfying assignments of a Boolean formula is hard in general, but it can be approximated efficiently via state-of-the-art PAC counters with access to an NP-oracle. In practice, an NP-oracle is a SAT solver capable of handling formulas with up to a million variables—orders of magnitude larger than typical network reliability instances.

2.3. Reducing network reliability to counting

Next we introduce the $\mathcal{K}$-RelNet formulation. Given propositional variables $S = (s_u)_{u \in V}$ and propositional variables $X = (x_e)_{e \in E}$, define:

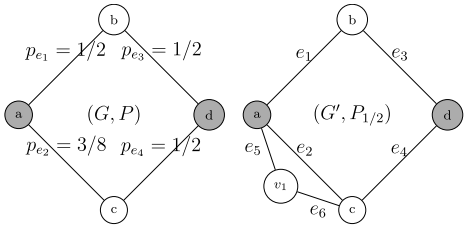
$$C_e = [(s_u \wedge x_e) \rightarrow s_v] \wedge [(s_v \wedge x_e) \rightarrow s_u], \forall\, e \in E, \tag{4}$$

$$F_{\mathcal{K}} = \exists\, S[\psi(X, S)] = \exists\, S\left[\left(\bigvee_{j \in \mathcal{K}} s_j\right) \wedge \left(\bigvee_{k \in \mathcal{K}} \neg\, s_k\right) \wedge \bigwedge_{e \in E} C_e\right], \tag{5}$$

where in Eq. (4), each edge $e \in E$ has end vertices $u, v \in V$. Propositional edge variable x_e encodes the state of edge $e \in E$, such that x_e is true iff e is not failed, which is consistent with the representation of a realization of the stochastic graph $G(P)$ introduced earlier. An example of $F_{\mathcal{K}}$ is given in Fig. 1b and 1c. Note that $F_{\mathcal{K}}$ is a Σ_1^1 formula,[4] and we define its associated set of satisfying assignments as $R_{F_{\mathcal{K}}} = \{X \in \Omega | (\exists S)\psi(X, S) = 1\}$, such that $\#F_{\mathcal{K}} = |R_{F_{\mathcal{K}}}|$. Also, recall that the notation for the complement of set Θ is $\overline{\Theta}$. The next Lemma proves the core result of our reduction.

Lemma 1. *For a graph $G = (V, E, \mathcal{K})$, edge failure probabilities $P = (p_e)_{e \in E}$, and $F_{\mathcal{K}}$ and Ω_f as defined above, we have $\#F_{\mathcal{K}} = |\Omega_f|$. Moreover, for $P_{1/2} = (1/2)_{e \in E}$, we have*

$$u_G(P_{1/2}) = \frac{\#F_{\mathcal{K}}}{2^{|E|}}.$$



(a) Weighted instance. (b) Unweighted instance.

$C_{e_1} = (s_a \wedge x_{e_1} \rightarrow s_b) \wedge (s_b \wedge x_{e_1} \rightarrow s_a)$, $C_{e_2} = (s_a \wedge x_{e_2} \rightarrow s_c) \wedge (s_c \wedge x_{e_2} \rightarrow s_a)$.
$C_{e_3} = (s_b \wedge x_{e_3} \rightarrow s_d) \wedge (s_d \wedge x_{e_3} \rightarrow s_b)$, $C_{e_4} = (s_c \wedge x_{e_4} \rightarrow s_d) \wedge (s_d \wedge x_{e_4} \rightarrow s_c)$.
$C_{e_5} = (s_a \wedge x_{e_5} \rightarrow s_{v_1}) \wedge (s_{v_1} \wedge x_{e_5} \rightarrow s_a)$, $C_{e_6} = (s_{v_1} \wedge x_{e_6} \rightarrow s_c) \wedge (s_c \wedge x_{e_6} \rightarrow s_{v_1})$.
$S = \{s_a, s_b, s_c, s_d, s_v\}$, $F_{\mathcal{K}} = \exists S\big((s_a \vee s_d) \wedge (\neg s_a \vee \neg s_d) \wedge \bigwedge_{i=1}^{6} C_{e_i}\big)$.
$\#F_{\mathcal{K}} = 33$, $u_G(P) = \#F_{\mathcal{K}}/2^{|E'|} = 33/64$.

(c) Terms of Σ_1^1 formula $F_{\mathcal{K}}$ and exact counting calculations.

Fig. 1. $\mathcal{K}$-RelNet example with $\mathcal{K} = \{a, b\}$. (a) Original instance, (b) its reduction to $p_e = 1/2$, $\forall e \in E'$, and (c) exact counting $\#F_{\mathcal{K}}$.

Proof. We use ideas from our previous work [20], which deals with the special case $|\mathcal{K}| = 2$. First, note that for sets A and B such that $|A| + |\overline{A}| = |B| + |\overline{B}|$, we have $|A| = |B|$ iff there is a bijective mapping from $\overline{A}$ to $\overline{B}$. Moreover, the number of unquantified variables in Eq. (5) is $|E|$, so we can establish the next equivalence between the number of distinct edge variable assignments and system states: $|R_{F_{\mathcal{K}}}| + |\overline{R_{F_{\mathcal{K}}}}| = |\Omega_f| + |\overline{\Omega_f}| = 2^{|E|}$. Next, we prove $X \in \overline{R_{F_{\mathcal{K}}}} \Leftrightarrow X \in \overline{\Omega_f}$, $\forall\, X \in \{0, 1\}^{|E|}$, via a bijective mapping.

1) Case $\overline{\Omega_f} \rightarrow \overline{R_{F_{\mathcal{K}}}}$: assume $X \in \overline{\Omega_f}$, i.e. $\phi(X) = 1$ or G is $\mathcal{K}$-connected. Next, we show that $X \in \overline{R_{F_{\mathcal{K}}}}$, i.e., $F_{\mathcal{K}}(X, S)$ evaluates to false for all possible assignments of variables S, due to Eqs. (4) and (5). We show this by way of contradiction. Assume there is an assignment $S \in \{0, 1\}^{|V|}$ such that $F_{\mathcal{K}}(X, S)$ is true. We deduce this happens iff (i) $\exists\, j, k \in \mathcal{K}$ such that $s_j \neq s_k$, from Eq. (5), and (ii) for every edge $e \in E$ with end-vertices $u, v \in V$ we have $s_v = 1$ (resp. $s_u = 1$) whenever x_e and s_u (resp. s_v) are equal to 1, due to clause C_e in Eq. (4). Without loss of generality, we satisfy condition (i) setting $s_j = 1$ and $s_k = 0$, with $j, k \in \mathcal{K}$. Recall $X \in \overline{\Omega_f}$, i.e. $\phi(X) = 1$, so there is a path $P' = \{j, \cdots, k\} \subseteq V$ connecting vertices $j, k \in \mathcal{K}$ and traversing edges $T \subseteq E$ such that $x_e = 1$, $\forall\, e \in T$. By iterating over constraints C_e, $\forall e \in T$, and since $s_j = 1$, we are forced to assign $s_i = 1$, $\forall\, i \in P'$, to satisfy condition (ii). This assignment results into $s_k = 1$, which contradicts condition (i) when we have set $s_k = 0$ at the beginning. Thus, an $S \in \{0, 1\}^{|V|}$ such that $F_{\mathcal{K}}(X, S)$ is true does not exists, and $X \in \overline{R_{F_{\mathcal{K}}}}$.

2) Case $\overline{R_{F_{\mathcal{K}}}} \rightarrow \overline{\Omega_f}$: assume $X \in \overline{R_{F_{\mathcal{K}}}}$, i.e. $F_{\mathcal{K}}(X, S)$ is false, to show that $X \in \overline{\Omega_f}$. Again, by way of contradiction, we assume $X \in \Omega_f$, i.e. $\phi(X) = 0$. However, from $\overline{R_{F_{\mathcal{K}}}} \rightarrow \overline{\Omega_f}$ and the transitivity property in the constraints of Eq. (4), we deduce that the set of edges $T = \{e \in E | x_e = 1\}$ connects every pair of vertices $i, j \in \mathcal{K}$, i.e. $\phi(X) = 1$ by definition of ϕ. Having reached a contradiction, we conclude $X \in \overline{\Omega_f}$.

Since we established a bijective mapping between $\overline{R_{F_{\mathcal{K}}}}$ and $\overline{\Omega_f}$, we conclude $\#F_{\mathcal{K}} = |\Omega_f|$. The last part of the lemma follows by noting that $\Pr(X) = 1/2^{|E|}$ when $P = P_{1/2}$, so that $u_G(P) = \sum_{X \in \Omega} (1 - \Phi(X)) \cdot \Pr(X) = |\Omega_f| \cdot 1/2^{|E|} = \#F_{\mathcal{K}}/2^{|E|}$. □

Now we generalize $u_G(P_{1/2}) = \#F_{\mathcal{K}}/2^{|E|}$ to arbitrary edge failure probabilities. To this end, we use a weighted to unweighted transformation [20].

2.4. Addressing arbitrary edge failure probabilities

The next definitions will be useful for stating our weighted-to-unweighted transformation. Let $0.b_1 \cdots b_m$, with $b_i \in \{0, 1\}$, be the binary representation of probability $q \in (0, 1)$, i.e. $q = \sum_{k=1}^{m} b_k/2^k$. Define z_k

[4] Use identity $(a \wedge b) \rightarrow c \equiv \neg a \vee \neg b \vee c$ for constraints C_e in Eq. (4).

($\bar{z}_k$) as the number of zeros (ones) in the first k decimal bits of the binary representation. Formally, $z_k = k - \sum_{i=1}^{k} b_i$ and $\bar{z}_k = k - z_k$, $\forall k \in L$, with $L = \{1, \cdots, m\}$. Moreover, for $V = \{v_0, \cdots, v_{z_m+1}\}$, define a function η: $L \to V \times V$ such that $\eta(k) = (v_{z_k-1}, v_{z_k})$ if $b_k = 0$, and $\eta(k) = (v_{z_k-1}, v_{z_m+1})$ otherwise. We will show that, for $E = \bigcup_{k\in L}\eta(k)$ and $\mathcal{K} = \{v_0, v_{z_m+1}\}$, $G(V, E, \mathcal{K})$ is a series-parallel graph such that $r_G(P_{1/2}) = q$. Thus, our weighted-to-unweighted transformation entails replacing every edge $e \in E$ with a reliability preserving series-parallel graph G^e where every edge fails with probability 1/2.

For example, from Fig. 1, the binary representation of $1 - p_{e_2} = 5/8$ is 0.101, so we have $m = 3$, $z_m = 1$, and $\bar{z}_m = 2$. Also, we replace edge e_2 with a series parallel graph G^{e_2} using the construction from above, which yields $V^{e_2} = \{v_0, v_1, v_2\}$, $E^{e_2} = \{(v_0, v_2), (v_0, v_1), (v_1, v_2)\}$, and terminal set $\mathcal{K}^{e_2} = \{v_0, v_2\}$. Since $u_{G^{e_2}}(P_{1/2}) = 3/8$, we replace e_2 by G^{e_2} as shown in Fig. 1, where $v_0 = a$ and $v_2 = c$, for consistency with the global labeling of the figure. The next lemma proves the correctness of this transformation.

Lemma 2. *Given probability $q = 0.b_1 \cdots b_m$ in binary form, graph $G = (V, E, \mathcal{K})$ such that $V = \{v_0, \cdots, v_{z_m+1}\}$, $E = \{\eta(1), \cdots, \eta(m)\}$ and $\mathcal{K} = \{v_0, v_{z_m+1}\}$, and edge failure probabilities $P_{1/2} = (1/2)_{e\in E}$, we have $r_G(P_{1/2}) = q$ and $|V| + |E| = z_m + 2 + m$.*

Proof. Define $G_k = (V_k, E_k)$, $\forall\, k \in L$, with $E_k = \{\eta(1), \cdots, \eta(k)\}$ and $V_k = \cup_{i=1}^{k}\{v_j\colon v_j \in \eta(i)\}$. Clearly, $V = V_m$ and $E = E_m$. The key observation is that G is a series-parallel graph and that we can enumerate all paths from v_0 to v_{z_m+1} in G. Let $k_1 = \min\{k \in L\colon b_{k_1} = 1\}$. Then, the edge set $E_{T_1} = E_{k_1}$ forms a path from v_0 to v_{z_m+1}, denoted T_1, with vertex sequence $(v_0, \cdots, v_{z_{k_1}}, v_{z_m+1})$, size $|E_{T_1}| = z_{k_1} + 1$, and $\Pr(T_1) = 1/2^{z_{k_1}+1}$. Next, for k_2 the second smallest element of L such that $b_{k_2} = 1$, G_{k_2} contains a total of two paths, T_1 and T_2, with T_1 as before and $E_{T_2} = E_{k_2} \setminus \{(v_{z_{k_1}}, v_{m+1})\}$ of size $z_{k_2} + 1$. Also, $E_{k_2} = E_{T_1} \cup E_{T_2}$ and $E_{T_1} \cap E_{T_2} = E_{T_1} \setminus \{(v_{z_{k_1}}, v_{m+1})\}$. Thus, the event $\overline{T}_1 T_2$ happens iff edge $(v_{z_{k_1}}, v_{m+1})$ fails and edges in E_{T_2} do not fail, letting us write $\Pr(\overline{T}_1 T_2) = 1/2 \cdot 1/2^{z_{k_2}+1}$. For k_j the j-th smallest element of L such that $b_{k_j} = 1$, G_{k_j} has a total of $j = \bar{z}_{k_j}$ paths, with $E_{T_j} = E_{k_j} \setminus \cup_{i=1}^{j-1}\{(v_{z_{k_i}}, v_{m+1})\}$, $|E_{T_j}| = z_{k_k} + 1$, and $E_{kj} = \cup_{i=1}^{j} E_{T_i}$. Furthermore, event $\overline{T}_1 \cdots \overline{T}_{j-1} T_j$ happens iff edges in $\cup_{i=1}^{j-1}\{(v_{z_{k_i}}, v_{m+1})\}$ fail and edges in E_{T_j} do not fail. Thus, $\Pr(\overline{T}_1 \cdots \overline{T}_{j-1} T_j) = 1/2^{\bar{z}_{kj}-1} \cdot 1/2^{z_{kj}+1} = 1/2^{k_j}$. This leads to $r_G(P) = \Pr(T_1) + \Pr(\overline{T}_1 T_2) + \cdots + \Pr(\overline{T}_1 \cdots \overline{T}_{\bar{z}_m - 1} S_{\bar{z}_m}) = \sum_{i=1}^{\bar{z}_m} 1/2^{k_i}$. Rewriting the summation over all $k \in L$ yields $r_G(P) = \sum_{k=1}^{m} b_k/2^k$, which is the decimal form of $q = 0.b_k \cdots b_m$. Furthermore, $|V| = z_m + 2$ and $|E| = m$ from their definitions. □

Now we leverage Lemma 2 to introduce our general counting-based algorithm for the $\mathcal{K}$-terminal reliability problem.

2.5. The new algorithm: $\mathcal{K}$-RelNet

$\mathcal{K}$-RelNet is presented in Algorithm 1. Theorem 3 proves its correctness. Fig. 1 illustrates the exact version beginning with the reduction to failure probabilities of 1/2, and rounding up with the construction of $F_{\mathcal{K}}$ and exact counting of its satisfying assignments. In Algorithm 1, however, we use an approximate counter giving (ϵ, δ) guarantees [16].

Theorem 3. *Given an instance (G, P) of the $\mathcal{K}$-terminal reliability problem and M defined as in Algorithm 1:*

$u_G(P) = \#F_{\mathcal{K}}/2^M$.

Proof. The proof follows directly from Lemmas 1 and 2. First, note that the transformation in step 1 of $\mathcal{K}$-RelNet outputs an instance $(G', P_{1/2})$ so that $u_G(P) = u_{G'}(P_{1/2})$, where $P_{1/2}$ denote edges in E' that fail with probability 1/2 (Lemma 2). Then, step 2 takes G' to output $F_{\mathcal{K}}$ such that $u_{G'}(P_{1/2}) = |R_{F_{\mathcal{K}}}|/2^M$ (Lemma 1). Finally, $u_G(P) = |R_{F_{\mathcal{K}}}|/2^M$. □

Steps 1–2 run in polynomial time on the size of (G, P). Step 3 invokes ApproxMC2 [15] to approximate $\#F_{\mathcal{K}}$. In turn, ApproxMC2 has access to a SAT-oracle, running in polynomial time on $\log 1/\delta$, $1/\epsilon$, and $|F_{\mathcal{K}}|$. Thus, relative to a SAT-oracle, $\mathcal{K}$-RelNet approximates $u_G(P)$ with (ϵ, δ) guarantees in the FPRAS theoretical sense. Also, we note that ApproxMC2's (ϵ, δ) guarantees are for the multiplicative error $\Pr(1/(1+\epsilon)u_G(P) \le \hat{u}_G(P) \le (1+\epsilon)u_G(P)) \ge 1 - \delta$ [15]. This is a tighter error constraint than the relative error of Eq. (3), as one can show that $1 - \epsilon \le 1/(1+\epsilon)$ for $\epsilon \in (0, 1)$. Thus, if an approximation method satisfies the multiplicative error guarantees, then it also satisfies the relative error guarantees. The converse is not true, and herein we will omit this advantage of $\mathcal{K}$-RelNet over other methods for ease of comparison. Moreover, a SAT-oracle is a SAT-solver able to answer satisfiability queries with up to a million variables in practice. $F_{\mathcal{K}}$ has $|V'| + |E'|$ variables, thus facilitating implementation in practice. Given $\mathcal{K}$-RelNet's theoretical guarantees, we now provide context relative to other existing methods, to then perform computational experiments verifying its performance in practice.

3. Context relative to competitive methods

This section briefly contextualizes our work relative to competitive techniques for network reliability evaluation, so as to facilitate the comparative analyses in Section 4. We arrange methods into three groups: exact or bounds, guarantee-less simulation, and probably approximately correct (PAC).

3.1. Group (i): exact or bounds

Network reliability belongs to the computational complexity class #P-complete, which is largely believed to be intractable. This means that the task of computing $u_G(P)$ efficiently is seemingly hopeless. While of limited application, the most popular techniques in this group employ approaches such as state enumeration [22], direct decomposition [23], factoring [24], or compact data structures like binary-decision-diagrams (BDD) [3]. We refer the reader to the cited literature for a survey of exact methods [22,25].

The intractability of reliability problems motivates exploiting properties from graph theory. For example, in the case of bounded therewidth and degree, there are efficient algorithms available [3,4]. Another promising family of methods issues fast converging bounds [23,26], an approach that demonstrates practical performance even in earthquake engineering applications [27], and that is applicable beyond connectivity-based reliability as part of the more general state-space-partition principle [28,29].

Input: Instance (G, P) and (ϵ, δ)-parameters.
Output: PAC estimate $\hat{u}_G(P)$.
1: Construct $G' = (V', E', \mathcal{K})$ replacing every edge $e \in E$ by G^e such that $1 - p_e = 0.b_1 \cdots b_{m_e}$ and with $r_{G^e}(P_{1/2}) = 1 - p_e$ (Lemma 2).
2: Let $M = \sum_{e\in E} m_e = |E'|$, and construct $F_{\mathcal{K}}$ using G' from Eq. 5.
3: Invoke ApproxMC2, a hashing-based counting technique [16], to compute $\widehat{\#F_{\mathcal{K}}}$, an approximation of $\#F_{\mathcal{K}}$ with (ϵ, δ) guarantees.
$\hat{u}_G(P) \leftarrow \widehat{\#F_{\mathcal{K}}}/2^{|M|}$

Algorithm 1. $\mathcal{K}$-RelNet.

3.2. Group (ii): guarantee-less simulation

When exact methods fail to scale, guarantee-less simulations have found wide applicability. In the context of unbiased estimators,[5] a key property is the relative variance σ_Y^2/μ_Y^2, with Y a randomized Monte Carlo procedure such that $E[Y] = u_G(P)$. From Theorem 6 (Appendix), we know that should a method verify the bounded relative variance (BRV) property, i.e., $\sigma_Y^2/\mu_Y^2 \leq C$ for C some constant, then an efficient (ϵ, δ) approximation is guaranteed with a sample size of $N = O(\epsilon^{-2}\log 1/\delta)$. While certain methods verify the BRV property, the value of C is typically unknown for general instances of the $\mathcal{K}$-terminal reliability problem, and thus the central limit theorem is often invoked for drawing confidence intervals despite known caveats [18]. Some techniques verifying the BRV property include the permutation Monte Carlo-based Lomonosov's Turnip (LT) [8] and its sequential splitting extension, the Split-Turnip (ST) [13], and the importance sampling variants of the recursive variance reduction (RVR) algorithm [30]. They significantly outperform the crude Monte Carlo (CMC) method in the rare event setting, with RVR even displaying the VRV property in select instances, as evidenced in empirical evaluations.

As we noted, the number of samples in the crude Monte Carlo approach scales like $1/u_G(P)$, which can be problematic in highly-reliable systems. A more promising approach leverages the Markov Chain Monte Carlo method and the product estimator [31,32], where the small $u_G(P)$ estimation is bypassed by estimating the product of larger quantities. Significantly, the sample size roughly scales like $\log 1/u_G(P)$ [33]. The product estimator is popularly referred to as multilevel splitting as it has independently appeared in other disciplines [34–36], and even more recently in the civil and mechanical engineering fields under the name of subset simulation [37]. In the case of network reliability, the latent variable formulation by Botev et al. [9], termed generalized splitting (GS), delivers unbiased estimates of $u_G(P)$. The similar approach by Zuev et al. [10] is not readily applicable to the $\mathcal{K}$-terminal reliability and delivers biased samples, which makes it problematic when one wants to rigorously assess confidence.

3.3. Group (iii): PAC methods

In a breakthrough paper, Karger gave the first efficient approximation for the all-terminal network unreliability problem [14]. However, Karger's algorithm is not always practical despite recent improvement [38]. Also, unlike $\mathcal{K}$-RelNet, Karger's algorithm is not readily applicable to the more general $\mathcal{K}$-terminal network reliability problem.

Besides our network reliability PAC approximation technique, $\mathcal{K}$-RelNet, which is specialized to the $\mathcal{K}$-terminal reliability problem, there are general Monte Carlo sampling schemes that deliver (ϵ, δ) guarantees. The reminder of this subsection highlights relevant methods that are readily implementable in Monte Carlo-based network reliability computations.

Denoting Y the random samples produced by unbiased sampling-based estimators, traditional simulation approaches take the average of i.i.d. samples of Y. Such estimators can be integrated into optimal Monte Carlo simulation (OMCS) algorithms [39]. An algorithm A is said to be optimal (up to a constant factor) when its sample size N_A is not proportionally larger in expectation than the sample size N_B of any other algorithm B that is also an (ϵ, δ) randomized approximation of μ_Y, and that has access to the same information as A, i.e., $E[N_A] \leq c \cdot E[N_B]$ with c a universal constant.

A simple and general purpose black box algorithm to approximate $u_G(P)$ with PAC guarantees is the *Stopping Rule Algorithm* (*SRA*) introduced by Dagum et al. [39]. The convergence properties of *SRA* were shown through the theory of martingales and its implementation is straightforward (Algorithm 2).

Even though *SRA* is optimal up to a constant factor for RVs with support {0, 1}, a different algorithm and analysis leads to the *Gamma Bernoulli Approximation Scheme* (*GBAS*) [40], which improves the expected sample size by a constant factor over *SRA* and demonstrates superior performance in practice due to improved lower order terms in its guarantees. *GBAS* has the additional advantage with respect to *SRA* of being unbiased, and it is relatively simple to implement. The core idea of *GBAS* is to construct a RV such that its relative error probability distribution is known. The procedure is shown in Algorithm 3, where I is the indicator function, Unif(0, 1) is a random draw from the uniform distribution bounded in [0,1], and Exp(1) is a random draw from an exponential distribution with parameter $\lambda = 1$. Also, Algorithm 3 requires parameter k, which is set as the smallest value that guarantees $\delta \geq \Pr(\mu_Y/\hat{\mu}_Y < (1+\epsilon)^2 \text{ or } \mu_Y/\hat{\mu}_Y > (1-\epsilon)^2)$ with $\mu_Y/\hat{\mu}_Y \sim \text{Gamma}(k, k-1)$ [40]. In practice, values of k for relevant (ϵ, δ) pairs can be tabulated. Alternatively, if one can evaluate the cumulative density function (cdf) of a Gamma distribution, galloping search can be used to find the optimal value of k with logarithmic overhead (on the number of cdf evaluations).

Note that *SRA* and *GBAS* give PAC estimates with optimal expected number of samples for RVs with support {0, 1}, yet they disregard the variance reduction properties of more advanced techniques. Thus, one can ponder, is there a way to exploit a randomized procedure Y such that $\sigma_Y \ll \sigma_{Y^{CMC}}$ in the context of OMCS? The *Approximation Algorithm* ($\mathcal{AA}$), introduced by Dagum et al. [39], and based on sequential analysis [41], gives a partially favorable answer. In particular, steps 1 and 2 of Algorithm 4 are trial experiments that give rough estimates of μ_Y and σ_Y^2/μ_Y^2, respectively. Then, step 3 is the actual experiment that outputs $\hat{u}_G(P)$ with PAC guarantees. $\mathcal{AA}$ assumes $Y \in [0, 1]$, and it was shown to be optimal up to a constant factor.

The downside of $\mathcal{AA}$, or any OMCS algorithm as $\mathcal{AA}$ is optimal, is that it requires in expectation $N_{\mathcal{AA}} = O(\max\{\sigma_Y^2/\mu_Y^2, \epsilon/\mu_Y\}\cdot\epsilon^{-2}\ln 1/\delta)$ samples. Thus, despite considering the relative variance σ_Y^2/μ_Y^2, OMCS algorithms become impractical in the rare-event regime. For example, consider the case in which edge failure probabilities tend to zero and $1/\mu_Y$ goes to infinity. If a technique delivers Y that meets the BRV property, i.e., $\sigma_Y^2/\mu_Y^2 \leq C$ for C some constant, then, from Theorem 6 (Appendix), we know a sample of $N = O(\epsilon^{-2}\ln 1/\delta)$ suffices, meanwhile $N_{\mathcal{AA}} \to \infty$.

We will use *GBAS* for CMC with (ϵ, δ) guarantees, and use $\mathcal{AA}$, given its generality, to turn various existing techniques into PAC methods themselves. For $\mathcal{AA}$, note that the rough estimate $\hat{\mu}_Y$ in step 1 is computed using Y^{CMC} as it is the cheapest, but from step 2 and on, the estimator that is intended to be tested is used, with the reported runtime being that of step 3 to measure variance reduction and runtime without trial experiments using CMC.

4. Computational experiments

A fair way to compare methods is to test them against challenging benchmarks and quantify empirical measures of performance relative to their theoretical promise. We take this approach to test $\mathcal{K}$-RelNet alongside competitive methods. The following subsections describe our experimental setting, list implemented methods, and show their application to various benchmarks.

4.1. Implemented estimation methods

Table 1 lists reliability evaluation methods that we consider in our numerical experiments. Exact methods run until giving an exact estimate or best bounds until timeout. Each guarantee-less simulation method uses a custom number of samples N that depends on the shared parameter N_S (Table 1). This practice, borrowed from Vaisman

[5] The quality of a guarantee-less method being unbiased is key, as boosting confidence by means of repeating experiments leveraging the central limit theorem would lack justification otherwise.

Input: $\epsilon, \delta \in (0,1)$ and random variable Y.
Output: Estimate $\hat{u}_G(P)$ with PAC guarantees.
Let $\{Y_i\}$ be a set of i.i.d samples of Y.
Compute constants $\Upsilon = 4(e-2)\log(2/\delta)1/\epsilon^2$, $\Upsilon_1 = 1 + (1+\epsilon)\cdot\Upsilon$.
Initialize $S \leftarrow 0$, $N \leftarrow 0$.
while $(S < \Upsilon_1)$ **do:** $N \leftarrow N+1$, $S \leftarrow S + Y_N$.
$\hat{u}_G(P) \leftarrow \Upsilon_1/N$

Algorithm 2. Stopping Rule Algorithm (*SRA*) [39].

Input: k parameter.
Output: Estimate $\hat{u}_G(P)$ with PAC guarantees.
Let $\{Y_i\}$ be a set of independent samples.
Initialize $S \leftarrow 0$, $R \leftarrow 0$, $N \leftarrow 0$.
while $(S \neq k)$ **do**
  $N \leftarrow N+1$, $B \leftarrow I(\text{Unif}(0,1) \leq Y_N)$
  $S \leftarrow S + B$, $R \leftarrow R + \text{Exp}(1)$
end while
$\hat{u}_G(P) \leftarrow (k-1)/R$

Algorithm 3. Gamma Bernoulli Approximation Scheme (*GBAS*) [40].

Input: (ϵ, δ)-parameters.
Output: Estimate $\hat{u}_G(P)$ with PAC guarantees.
Let $\{Y_i\}$ and $\{Y'_i\}$ be two sets of independent samples of Y.
1: $\epsilon' \leftarrow \min\{1/2, \sqrt{\epsilon}\}$, $\delta' \leftarrow \delta/3$
  $\hat{\mu}_Y \leftarrow SRA(\epsilon', \delta')$
2: $\Upsilon \leftarrow 4(e-2)\log(2/\delta)1/\epsilon^2$
  $\Upsilon_2 \leftarrow 2(1+\sqrt{\epsilon})(1+2\sqrt{\epsilon})(1+\ln(3/2)/\ln(2/\delta))\Upsilon$
  $N \leftarrow \Upsilon_2 \cdot \epsilon/\hat{\mu}_Y$, $S \leftarrow 0$
  for $(i = 1, \ldots, N)$ **do**: $S \leftarrow S + (Y'_{2i-1} - Y'_{2i})^2/2$
  $\hat{r}^*_Y \leftarrow \max\{S/N, \epsilon\hat{\mu}_Y\}/\hat{\mu}^2_Y$
3: $N_{\mathcal{AA}} \leftarrow \Upsilon_2 \cdot \hat{r}^*_Y$
  for $(i = 1, \ldots, N_{\mathcal{AA}})$ **do**: $S \leftarrow S + Y_i$
  $\hat{u}_G(P) \leftarrow S/N$

Algorithm 4. The Approximation Algorithm ($\mathcal{AA}$) [39].

all methods in a Python prototype[6] for uniform comparability and ran all experiments in the same machine—a 3.60 GHz quad-core Intel i7-4790 processor with 32GB of main memory and each experiment was run on a single core.

4.2. Estimator performance measures

We use the next empirical measures to assess the performance of reliability estimation methods. Let $\hat{u}$ be an approximation of u. We measure the *observed multiplicative error* ϵ_o as $(\hat{u}-u)/u$ if $\hat{u} > u$, and $(\hat{u}-u)/\hat{u}$ otherwise. Also, for a fixed PAC-method, target relative error ϵ, and independent measures $\epsilon_o^{(1)}, \cdots, \epsilon_o^{(M)}$, we compute the *observed confidence* parameter δ_o as $1/M \cdot \sum_{i=1}^{M} \mathbb{I}(|\epsilon_o^{(i)}| \geq \epsilon)$. Satisfaction of (ϵ, δ) is guaranteed, but ϵ_o and δ_o can expose theoretical guarantees that are too conservative.

Furthermore, for guarantee-less sampling methods we measure ϵ_o but not δ_o, as these do not support confidence a priori. Thus, we use empirical measures of variance reduction to assess the desirability of sampling techniques over the canonical method (CMC). Let $\sigma^2_{Y^{CMC}} = \mu_Y \cdot (1-\mu_Y)/N$ be the variance associated to CMC, and let $\sigma^2_{Y^A}$ be the sample variance associated to method A. Clearly, $\sigma^2_{Y^{CMC}}/\sigma^2_{Y^A} > 1$ will favor A over CMC. However, this is not the only important consideration in practice. For respective CPU times $\tau_{Y^{CMC}}$ and τ_{Y^A}, a ratio $\tau_{Y^{CMC}}/\tau_{Y^A} < 1$ would imply a higher computational cost for A. To account for both, variance and CPU time, we use the *efficiency ratio*, defined as $\text{ER}(Y^A) = \left(\sigma^2_{Y^{CMC}}/\sigma^2_{Y^A}\right) \cdot (\tau_{Y^{CMC}}/\tau_{Y^A})$ [43]. In practice, when ER $(Y^A) < 1$, one prefers the more straightforward CMC. A similar measure in the literature is the *work normalized relative variance* [9], defined as $\text{wnrv}(Y) = \tau_Y \sigma^2_Y/\mu^2_Y$, which is related to the efficiency ratio via $\text{ER}(Y^A) = \text{wnrv}(Y^{CMC})/\text{wnrv}(Y^A)$. We prefer $\text{ER}(Y^A)$ over $\text{wnvr}(Y^A)$ as it is a measure of adequacy of A over CMC, informing users on whether they need to implement a more sophisticated method than CMC.[7]

Table 1
Methods used in computational experiments, and corresponding parameters.

Group	Methods	IDs	Parameters	Ref.
i	BDD-based network reliability	HLL	n/a	[3,42]
ii	Lomonosov's-Turnip	LT	$N = N_S$	[8]
	Sequential splitting Monte Carlo	ST	$B = 100$, $N = N_S/B$	[13]
	Generalized splitting	GS	$s = 2$, $N_0 = 10^3$, $N = N_S$	[9]
	Recursive variance reduction	RVR	$N = N_S/\binom{\lvert\mathcal{K}\rvert}{2}$	[30]
iii	Karger's 2-step Algorithm	K2Simple	ϵ, δ	[38]
	Optimal Monte Carlo simulation	*GBAS*, $\mathcal{AA}$	ϵ, δ	[39,40]
	Counting-based network unreliability	$\mathcal{K}$-RelNet	ϵ, δ	This paper

et al. [13], tries to account for the varying computational cost of samples among methods.

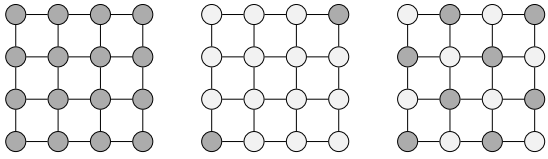
PAC algorithms $\mathcal{AA}$ or *GBAS* are used in combination with guarantee-less sampling methods to compare runtime given a target precision. For example, $GBAS(Y^{\text{CMC}})$ denotes Algorithm 3 when samples are drawn from the CMC estimator. Experiments with $\mathcal{AA}$ use $(\epsilon, \delta) = (0.2, 0.2)$. Experiments embedded in *GBAS* use two configurations: (0.2,0.2) and (0.2,0.05). $\mathcal{K}$-RelNet uses (0.8,0.2) to avoid time outs. As we will verify, in practice, PAC-methods can issue estimates with far better precision than the input theoretical (ϵ, δ)-guarantees.

To the best of our knowledge, methods in Table 1 are some of the best in their categories as evidenced in the literature. We implemented

The next subsections introduce the benchmarks we use and discuss

[6] RelNet's implementation can be found at https://github.com/meelgroup/RelNet.

[7] The ratio $\sigma^2_{Y^{CMC}}/\sigma^2_{Y^A}$ in the ER is also the ratio of the relative variances of Y^{CMC} and Y^A, shedding light on how many times larger (or smaller) the sample associated to CMC needs to be with respect to A from Theorem 6 (Appendix).



(a) All-Terminal (b) Two-Terminal (c) $\mathcal{K}$-Terminal

Fig. 2. Example of an $N \times N$ grid graph, with $N = 4$. Darkened nodes belong to the terminal set $\mathcal{K}$.

results. Also, in our benchmarks we consider sparse networks, i.e. $|E| = O(|V|)$, which resemble engineered systems.

4.3. Rectangular grid networks

We consider $N \times N$ square grids (Fig. 2) because they are irreducible (via series-parallel reductions) for $N > 2$, their tree-width is exactly N, and they can be grown arbitrarily large until exact methods fail to return an estimate. Also, failure probabilities can be varied to challenge simulation methods. Our goal is to increase N and vary failure probabilities uniformly to verify running time, scalability, and quality of approximation. We evaluate performance until methods fail to give a desirable answer. In particular, we consider values of N in the range 2 to 100, with the largest network being a 100×100 grid. Also, assume all edges fail with probability 2^{-i}, for $i \in \{1, 3, \cdots, 15\}$. Furthermore, we consider extreme cases of $\mathcal{K}$ (Fig. 2), namely, all-terminal and two-terminal reliability, and a $\mathcal{K}$-terminal case with terminal nodes distributed in a checkerboard pattern.

4.3.1. Exact calculations

For reference, we obtained exact unreliability calculations using the BDD-based method by Hardy, Lucet, and Limnios [3], herein termed HLL due to its authors. We computed $u_G(P)$ for $N = 2, .,10$ and all values of p_e. Fig. 3 shows a subset of exact estimates (a and b) and exponential scaling of running time (c). Several other exact methods we studied and referenced in Section 3, were used, but HLL was the only one that managed to estimate $u_G(P)$ exactly for all $N \leq 10$. However, HLL became memory-wise more consuming for $N > 10$. Thus, if memory is the only concern, the state-space partition can be used instead to get anytime bounds on $u_G(P)$ at the expense of larger runtime, but storing at most $O(|E|)$ vectors $X \in \{0, 1\}^m$ simultaneously [28]. Next, we use these exact estimates to compute ϵ_o and ER for guarantee-less simulation methods, and to compute ϵ_o and δ_o for PAC methods.

4.3.2. Guarantee-less simulation methods

Fig. 4 shows values of ϵ_o for the case of two-terminal reliability and setting $N_S = 10^4$. Most values are below the $\epsilon_o = 0.2$ threshold. For RVR we observed values of ϵ_o in the order of the float-point precision for the largest values of i. We attribute this to the small number of cuts with maximum probability (2–4 in our case) that, together with the fact that RVR finds them all in the decomposition process, endows RVR with the VRE property in this case. Conversely, other methods do not rely as heavily on these small number of cuts.

Moreover, the CPU time varied among methods as shown in Fig. 5. The only method whose single sample computation is affected by the values of i is GS, consistent with the expected number of levels, which scales as $\log 1/u_G(P)$. However, matrix exponential operations for handling more cases of i added overhead in LT and ST; the sharp time increase from $N = 5$ to $N = 6$ is due to this operation, consistent with findings by Botev et al. [9]. Instead, RVR does not suffer from numerical issues and appears to verify the VRV property in this grid topology.

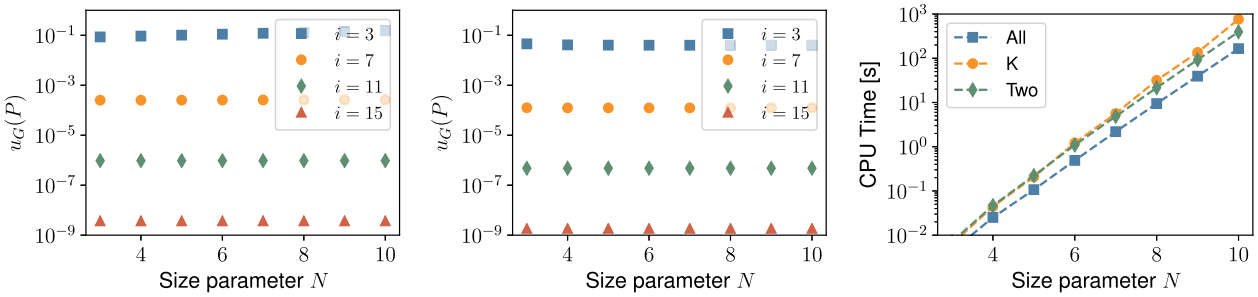
Also, to compare all methods in a uniform fashion we used the efficiency ratio (Fig. 6). Values of $\sigma^2_{Y^{\text{CMC}}}$ for computing the efficiency ratio are exact from HLL, and CPU time $\tau_{Y^{\text{CMC}}}$ is based on 10^4 samples. Estimates below the horizontal line are less reliable than those obtained with CMC for the same amount of CPU time. In particular, we note that for less rare failure probabilities ($2^{-7} \approx 0.008$) some methods fail to improve over CMC. Missing values for RVR show improvements above 10^7 in the efficiency ratio which, again, can be attributed to it meeting the VRE property in these benchmarks. Furthermore, an interesting result among simulation methods is that there is a downward trend in their efficiency ratio as N grows. Thus, we can construct an arbitrarily large squared grid for some N that will, ceteris paribus, yield an efficiency ratio below 1 in favor of CMC. We attribute this to the time complexity of CMC samples in sparse graphs, which can be computed in $O(|V|)$ time, whereas other techniques run in $O(|V|^2)$ time or worse. Thus, the larger the graph the far greater the cost per sample by more advanced techniques with respect to CMC.

4.3.3. Probably approximately correct (PAC) methods

Next, we embedded simulation methods in $\mathcal{AA}$, except CMC which was run using *GBAS* because the latter is optimal for Bernoulli RVs such as Y^{CMC}. Fig. 7 shows the runtime for methods embedded into $\mathcal{AA}$. We were able to feasibly compute PAC-estimates for edge failure probabilities of $2^{-5} \approx 0.03$ or larger across all methods. The approximation guarantees turned out to be rather conservative, obtaining far better precision in practice. Variance reduction through $\mathcal{AA}$ can only reduce sample size by a factor of $O(1/\epsilon)$ with respect to the Bernoulli case [i.e. $O(u_G(P)^{-1}\epsilon^{-2}\log\delta^{-1})$], thus PAC-estimates with advanced simulation methods using $\mathcal{AA}$ seem to be confined to cases where $u_G(P) \geq 0.005$ for the square grids benchmarks. However, conditioned on disruptive events such as natural disasters in which failure probabilities are larger, $\mathcal{AA}$ can deliver practical PAC-estimates.

On the other hand, $GBAS(Y^{CMC})$ turned out to be practical for more cases, and the analysis used by Huber [40] seems to be tight as evidenced by our estimates of δ_o (Fig. 8, a and b). Yet, as expected, the running time is heavily penalized by a factor $1/u_G(P)$ in the expected sample size as shown in Fig. 8(c).

Furthermore, we used $\mathcal{K}$-RelNet to approximate $u_G(P)$ in all cases of $\mathcal{K}$ thanks to our new developments. Fig. 9(b) shows runtimes as well as



(a) all-terminal reliability. (b) two-terminal reliability. (c) all cases.

Fig. 3. (a-b) Exact estimates of $u_G(P)$ and (c) CPU time using HLL for all-, two-, and $\mathcal{K}$-terminal cases.

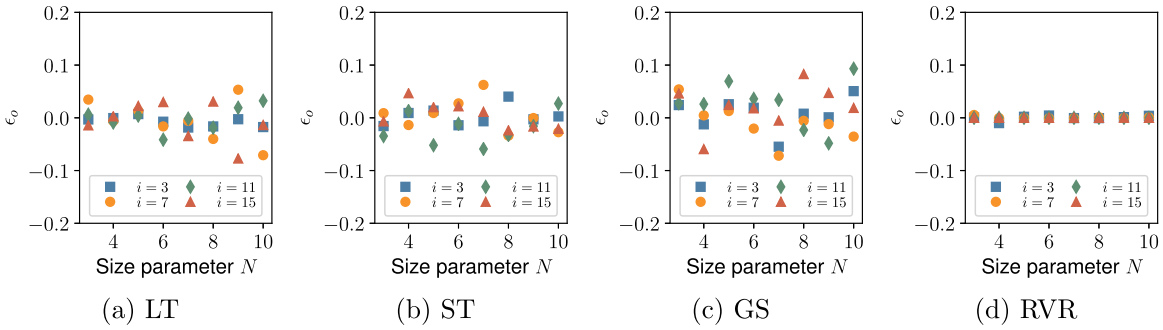


Fig. 4. Multiplicative error ϵ_o for guarantee-less simulation methods in the two-terminal reliability case.

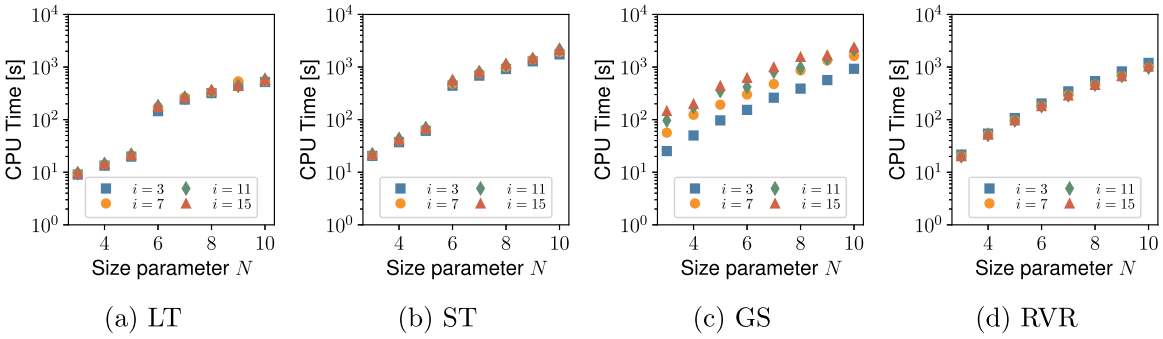


Fig. 5. CPU time for guarantee-less simulation methods in the two-terminal reliability case.

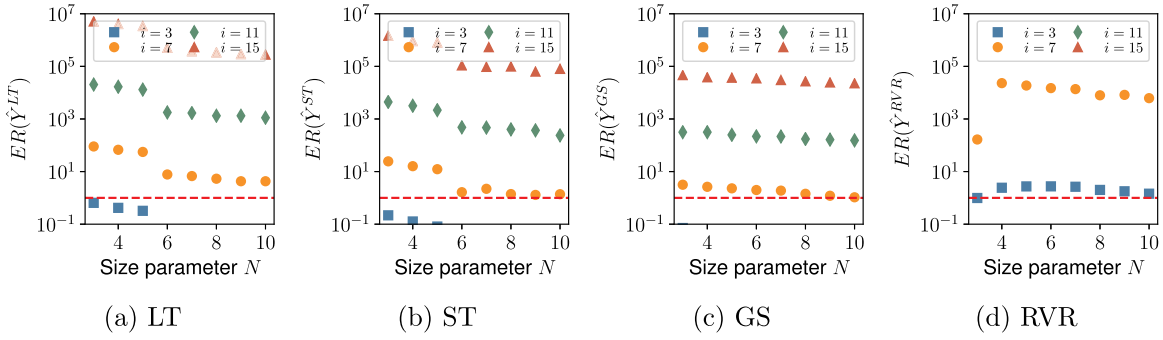


Fig. 6. ER for guarantee-less simulation methods in the two-terminal reliability case.

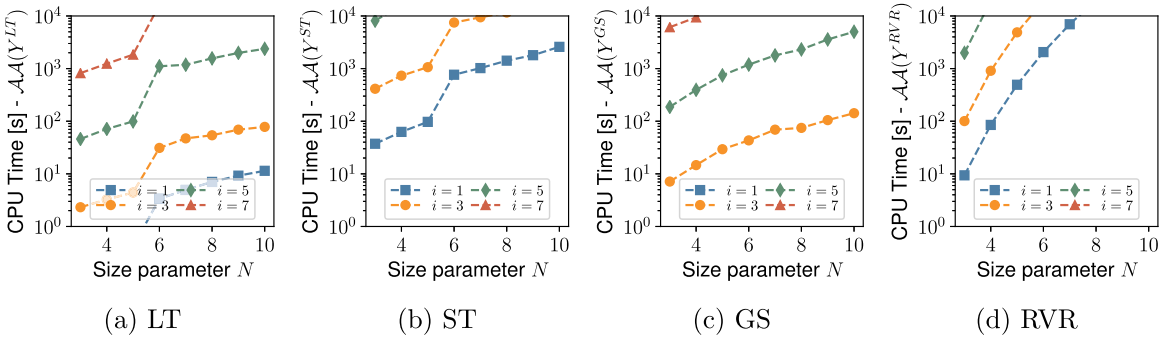


Fig. 7. CPU time for PAC-ized sampling methods via $\mathcal{AA}$ setting $\epsilon = \delta = 0.2$ (all-terminal reliability).

(δ_o, ϵ_o) values for edge failure probability cases of 2^{-1}, 2^{-3}, 2^{-5}. The weighted to unweighted transformation appears to be the current bottleneck as it considerably increases the number of extra variables in $F_{\mathcal{K}}$. However, note that, unlike K2Simple that is specialized for the all-terminal case [Fig. 9(b)], $\mathcal{K}$-RelNet is readily applicable to any $\mathcal{K}$-terminal reliability problem instance. Also, $\mathcal{K}$-RelNet is the only method that, due to its dependence on an external Oracle, can exploit on-going third-party developments, as constrained SAT and weighted model counting are very active areas of research.[8] Also, SAT-based methods are uniquely positioned to exploit breakthroughs in quantum hardware and support a possible quantum version of $\mathcal{K}$-RelNet [44].

[8] See past and ongoing competitions: http://satcompetition.org/.

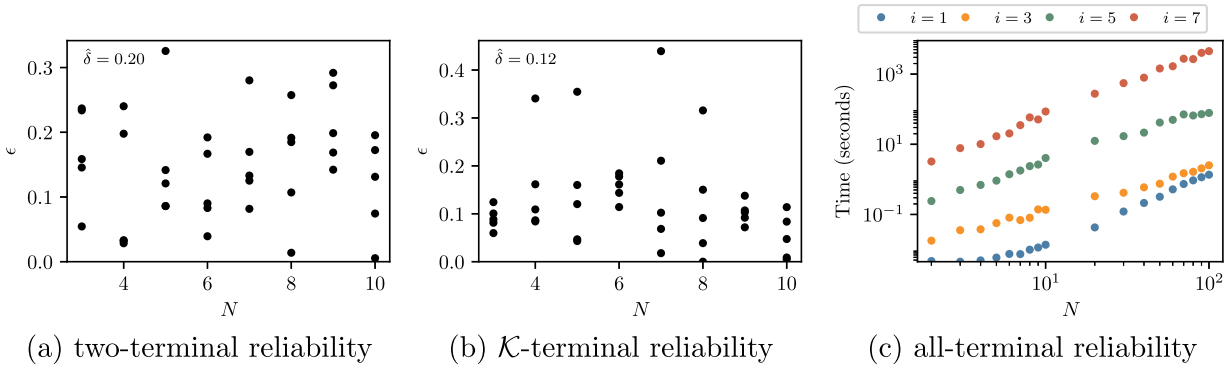


Fig. 8. (a-b) Multiplicative error for $GBAS(Y^{CMC})$ setting $\epsilon = \delta = 0.2$, and (c) respective running time for various sizes.

Furthermore, our experimental results suggest that the analysis of both, K2Simple and $\mathcal{K}$-RelNet, is not tight. This is observed by values of (ϵ_o, δ_o), which are far better than the theoretical input guarantees. This calls for further refinement in their theoretical analysis. Conversely, GBAS delivers practical guarantees that are much closer to the theoretical ones, as demonstrated in Figs. 8 and 10, where the target error can be exceeded still satisfying the target confidence overall.

The square grids gave us insight on the relative performance of reliability estimation methods. Next, we use a dataset of power transmission networks to test methods on instances with engineered system topologies.

4.4. U.S. power transmission networks

We consider a dataset with 58 power transmission networks in cities across the U.S. A summary discussion of their structural graph properties can be found elsewhere [45]. Also, we considered the two-terminal reliability problem. To test the robustness of methods, for each instance (G, P), we considered every possible $s, t \in V$ pair as a different experiment. Thus, totaling $\binom{|V|}{2}$ experiments per network instance. We used a single edge failure probability across experiments of $p_e = 2^{-3} = 0.125$ to keep overall computation time practical. Using HLL and preprocessing of networks, we were able to get exact estimates for some of the experiments. We used these to measure the observed multiplicative error ϵ_o when possible. Computational times are reported for all experiments, even if multiplicative error is unknown.

Fig. 11 shows PAC-estimates using $GBAS$. As expected, the variation in CPU time was proportional to $1/u_G(P)$. Furthermore, we used $\mathcal{K}$-RelNet to obtain PAC-estimates and observed consistent values of the multiplicative error (Fig. 12). In some instances, however, $\mathcal{K}$-RelNet failed to return an estimate before timeout. We also tested simulation methods setting $N_S = 10^3$. Despite the lack of guarantees they performed well in terms of ϵ_o and CPU time (Figs. 13 and 14, first 5 benchmarks for brevity). However, the efficiency ratio is reduced as the size of instances grows.

4.5. Analysis of results and outlook

Exact methods are advantageous when a topological property is known to be bounded. HLL proved useful not only for medium-sized grids ($N \times N = 100$), but also it was instrumental when computing exact estimates for many streamlined power transmission networks. Our research shows that methods exploiting bounded properties, together with practical upper bounds, deliver competitive exact calculations for many engineered systems. In power transmission networks, HLL was able to exploit their relatively small treewidth.

Among guarantee-less sampling methods, there are multiple paths for improvement. In the cases of LT and ST methods, even when the exponential matrix offers a reliable approach to compute the convolution of exponential random variables, numerically stable computations represent the main bottleneck of the algorithms, which appears to be useful mostly when both the network and edge failure probabilities are large. Thus, future research could devise ways to diagnose these numerical issues and fall-back to the exponential matrix only when needed, or use approximate integration (as in Gertsbakh et al. [46]), or use a more arithmetically robust algorithm (e.g. round-off algorithms for the sample variance [47]). Moreover, GS was competitive but its requirement to run a preliminary experiment with an arbitrary number of trajectories N_0 to define intermediate levels, and without a formal guidance on its values, can represent a practical barrier when there is no knowledge in the order of magnitude of $u_G(P)$. Future research could devise splitting mechanisms that use all samples towards the final experiments while retaining its unbiased properties. Finally, RVR was very competitive; however, we noted that (i) the number of terminals adds a considerable overhead in the number of calls to the minimum cut algorithm, and (ii) its performance is tied to the number of maximum probability cuts because larger cuts do not contribute meaningfully towards computing $u_G(P)$. Future work could use Karger's minimum cut approximation [48] and an adaptive truncation of the recursion found in the RVR estimator to address (i) and (ii), respectively. We are currently investigating this very issue and recognized the RVR estimator as

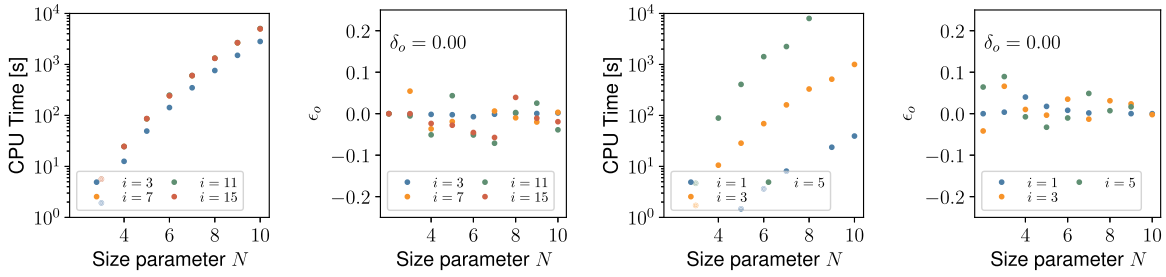
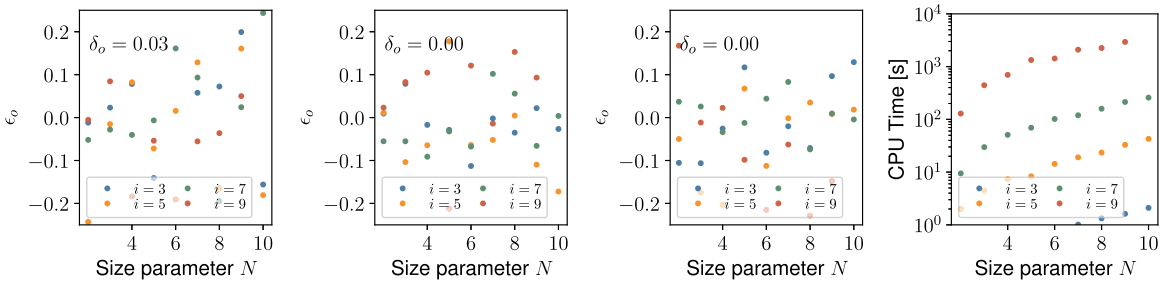
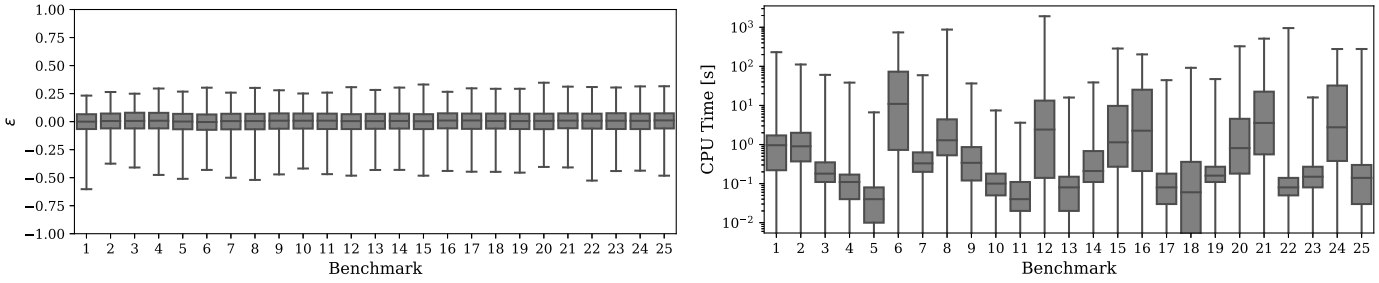


Fig. 9. $\mathcal{K}$-RelNet (a) and K2Simple (b) CPU time and ϵ_o values for $\mathcal{K}$-terminal reliability case.



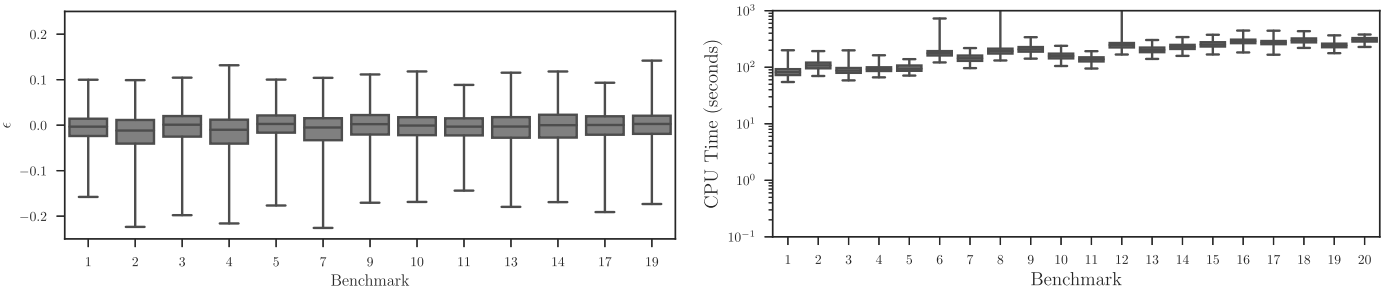
(a) All-terminal (b) Two-terminal (c) $\mathcal{K}$-terminal (d) Two-terminal

Fig. 10. ϵ_o values and CPU time for $GBAS(Y^{CMC})$ setting $(\epsilon, \delta) = (0.2, 0.05)$.



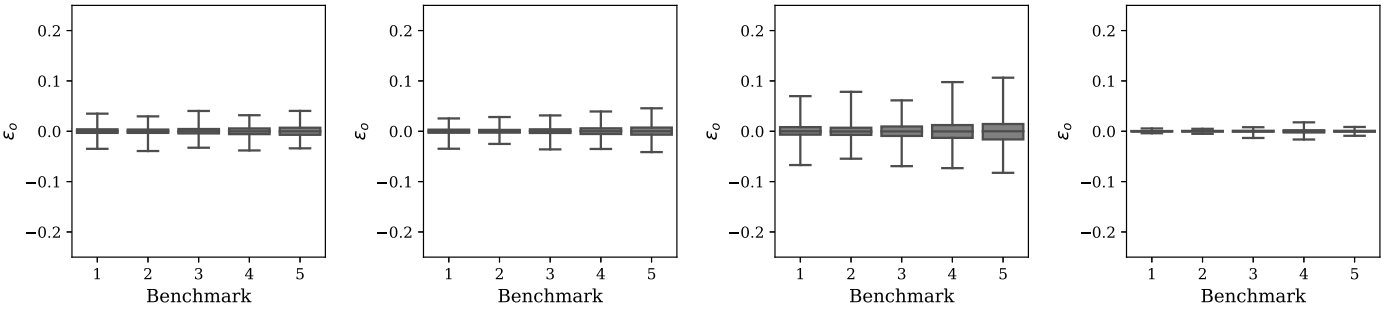
(a) Multiplicative error ϵ_o (b) Running time (seconds)

Fig. 11. Two-terminal reliability approximations in power grids using *GBAS* setting $\epsilon = \delta = 0.2$.



(a) Multiplicative error ϵ_o (b) Running time (seconds)

Fig. 12. Two-terminal reliability approximations in power grids using $\mathcal{K}$-RelNet with (0.8, 0.2).



(a) LT (b) ST (c) GS (d) RVR

Fig. 13. Multiplicative error in power grids for guarantee-less methods setting $N_S = 10^3$.

an special, yet randomized, case of state-space partition algorithms [28].

Among PAC-methods, we found GBAS to be tight in its theoretical analysis and competitive in practice. Outside the rare-event regime, since very small failure probabilities increase runtime, we contend that the usage of PAC algorithms such as GBAS would benefit the reliability and system safety community as they give exact confidence intervals without the need of asymptotic assumptions and arbitrary choices on the number of samples and replications. Karger's newly suggested algorithms demonstrated practical performance even in the rare-event regime, yet it appears that their theoretic guarantees are still too conservative. Equipping K2Simple with GBAS at the first recursion level

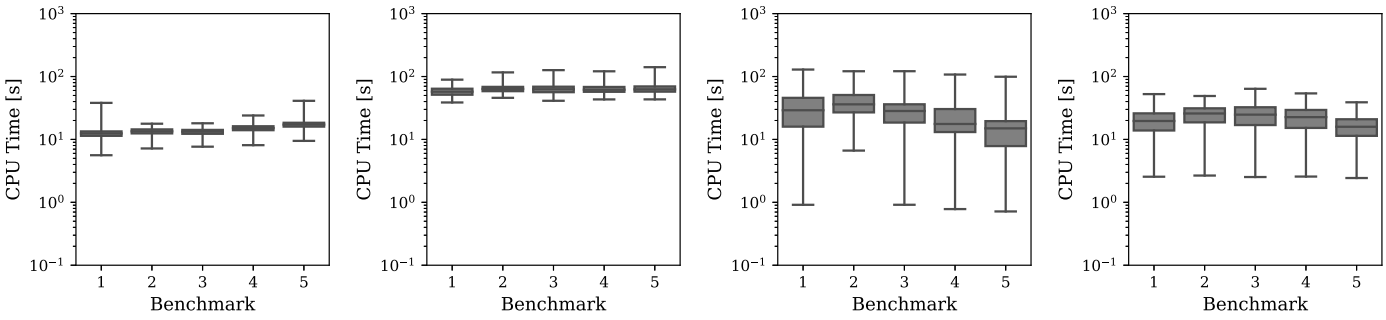


Fig. 14. Running time for simulation methods setting $N_S = 10^3$.

would instantly yield a faster algorithm for non-rare failure probabilities. However, the challenge of proving tighter bounds on the relative variance for the case of small failure probabilities remains. The same argument on theoretic guarantees being too conservative extends to $\mathcal{K}$-RelNet, which cannot be set too tight in practice. But we expect $\mathcal{K}$-RelNet to gain additional competitiveness as orthogonal advances in approximate weighted model counting continue to accrue. $\mathcal{K}$-RelNet remains competitive in the non rare-event regime, delivering rigorous PAC-guarantees for the $\mathcal{K}$-terminal reliability problem. Also, its SAT-based formulation makes it uniquely suitable for quantum algorithmic developments, at a time when major technological developers, such as IBM, Google, Intel, etc., are increasing their investment on quantum hardware [49].

5. Conclusions and future work

We introduced a new logic-based method for the $\mathcal{K}$-terminal reliability problem, $\mathcal{K}$-RelNet, which offers rigorous guarantees on the quality of its approximations. We examined this method relative to several other competitive approaches. For non-exact methods we emphasized desired relative variance properties: bounded by a polynomial on the size of the input instance [fully polynomial randomized approximation scheme (FPRAS)], bounded by a constant (bounded relative variance), or tending to zero (vanishing relative variance). We also turned popular estimators in the literature into probably approximately correct (PAC) ones by embedding them into an optimal Monte Carlo algorithm, and showed their practical performance using a set of benchmarks.

Contrasting with non-exact methods, our tool, $\mathcal{K}$-RelNet, is the first approximation of the $\mathcal{K}$-terminal reliability problem, giving strong performance guarantees in the FPRAS sense (relative to a SAT-oracle). Also, $\mathcal{K}$-RelNet gives rigorous multiplicative error guarantees, which are more conservative than relative error guarantees. However, its performance in practice remains constrained to not too small edge failure probabilities (≈ 0.1), which remains practical when conditioned on catastrophic hazard events. Thus, our future work will pursue more efficient encoding and solution approaches, especially when edge failure probabilities become smaller. Moreover, promising advances in approximate model counting and SAT solvers will render $\mathcal{K}$-RelNet more efficient over time, given its reliance on SAT oracles.

Embedding estimators with desired relative variance properties into PAC methods proved to be an effective strategy to bridge the gap between existing engineering reliability work and principled approximations, but only when failure probabilities are not rare. Despite this relative success, the strategy becomes impractical when $u_G(P)$ approaches zero. Thus, future research can address these issues in two fronts: (i) establishing parameterized upper bounds on the relative variance of new and previous estimators when they exist, and (ii) develop new PAC-methods with faster convergence guarantees than those of the canonical Monte Carlo approach.

Overall, PAC-estimation is a promising yet developing approach to system reliability estimation. Beyond the $\mathcal{K}$-terminal reliability problem, PAC methods can be used much more frequently as an alternative to the less rigorous—albeit pervasive—empirical methods that study variance through replications and asymptotic assumptions that appeal to the central limit theorem. In fact, methods such as *GBAS* deliver exact confidence intervals using all samples at the user's disposal. In future work, the authors will explore general purpose PAC-methods that can be employed in the rare-event regime, developing a unified framework to conduct reliability assessments with improved knowledge of uncertainties and further promote engineering resilience and align it with the measurement sciences.

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). This research is supported in part by the National Research Foundation Singapore under its AI Singapore Programme AISG-RP-2018-005 and NUS ODPRT Grant R-252-000-685-133.

Appendix

Proof of sample size large enough to deliver (ϵ, δ)*-guarantees (Theorem 6)*

The next two lemmas are useful towards proving Theorem 6. Lemma 4 shows how the variance in a Monte Carlo estimator reduces as a function of the number of samples.

Lemma 4. *For Y a random variable with mean μ_Y and variance σ_Y^2, define Y_n as follows:*

$$Y_n = \frac{1}{n}\sum_{i=1}^{n} Y_i,$$

with each $Y_i \sim Y$ and $n \in \mathbb{N}^+$. Then, we have that $\sigma_{Y_n}^2 = \frac{1}{n}\sigma_Y^2$.

Proof. The variance of Y_n is:

$$\begin{aligned}\sigma^2_{Y_n} &= E[(Y_n - \mu_{Y_n})^2] \\ &= E[Y_n^2] - \mu^2_{Y_n}.\end{aligned}$$

Note that $\mu_{Y_n} = \mu_Y$, and using the property of linearity in the expectation operator, write:

$$\begin{aligned}\sigma^2_{Y_n} &= \frac{1}{n^2}\sum_{i,j}^{n} E\left[Y_i Y_j\right] - \mu_Y^2 \\ &= \frac{1}{n^2}\left(\sum_{i}^{n} E[Y_i^2] + \sum_{i,j:i\neq j}^{n} E\left[Y_i Y_j\right]\right) - \mu_Y^2.\end{aligned}$$

Now, recall that every Y_i is i.i.d as Y and $\sigma_Y^2 = E[Y^2] - \mu_Y^2$. Then,

$$\begin{aligned}\sigma^2_{Y_n} &= \frac{1}{n^2}(n\cdot E[Y^2] + n(n-1)\cdot\mu_Y^2) - \mu_Y^2 \\ &= \frac{1}{n}(E[Y^2] + (n-1)\cdot\mu_Y^2 - n\cdot\mu_Y^2) \\ &= \frac{1}{n}(E[Y^2] - \mu_Y^2) \\ &= \frac{1}{n}\sigma_Y^2.\end{aligned}$$

□

Lemma 5 shows the link between the number of repetitions of a experiment and the success probability of the majority of repetitions. We will use this argument for constructing a median-based estimate.

Lemma 5. *Let X be a Bernoulli random variable with success probability $s \in [0, 1]$. Define the random variable:*

$$X_r = \sum_{i=1}^{r} X_i,$$

with $r \in \mathbb{N}^+$. Then, the probability of at most $\lfloor r/2 \rfloor$ successes is:

$$\Pr(X_r \leq r/2) = \sum_{i=0}^{\lfloor r/2 \rfloor} \binom{r}{i} (s)^i (1-s)^{r-i}$$

Proof. The proof is straightforward if one realizes that X_r is a Binomial random variable with parameters s and r. The desired probability is the cumulative distribution function evaluated at $r/2$. □

Next, we are ready to prove Theorem 6.

Theorem 6. *For a random variable Y with mean μ_Y and variance σ_Y^2, and user specified parameters ϵ, $\delta \in (0, 1)$, it suffices to draw $O(\sigma_Y^2/\mu_Y^2 \epsilon^{-2} \log 1/\delta)$ i.i.d samples to compute an estimate $\bar{\mu}_Y$ such that:*

$$\Pr\left(\frac{|\bar{\mu}_Y - \mu_Y|}{\mu_Y} \geq \epsilon\right) \leq \delta$$

Proof. From the well known Markov (or Chebyshev) inequality, we can write:

$$\Pr(|Y - \mu_Y| \geq k) \leq \frac{\sigma_Y^2}{k^2}.$$

For our purposes, we let $k = \epsilon\mu_Y$ with positive μ_Y. Then, we write:

$$\Pr\left(\frac{|Y - \mu_Y|}{\mu_Y} \geq \epsilon\right) \leq \frac{\sigma_Y^2}{\epsilon^2\mu_Y^2}.$$

If we substitute Y by Y_n such that $n = \frac{\sigma_Y^2}{(1-s)\epsilon^2\mu_Y^2}$ (Lemma 4), then:

$$\Pr\left(\frac{|Y_n - \mu_Y|}{\mu_Y} \geq \epsilon\right) \leq \frac{\sigma_Y^2/n}{\epsilon^2\mu_Y^2} = 1 - s.$$

Since the experiment's success probability is at least s, we boost it up to $1 - \delta$ via Lemma 5. First, let $\bar{\mu}_Y$ be the median of r samples of Y_n. Then, note that estimate $\bar{\mu}_Y$ "fails"—lays outside the interval $\mu_Y(1 \pm \epsilon)$—if and only if $r/2$ or more samples lay outside $\mu_Y(1 \pm \epsilon)$. Thus, choosing $s \in (0.5, 1)$, the probability that $\bar{\mu}_Y$ fails is at most:

$$\begin{aligned}\sum_{i=0}^{\lfloor r/2\rfloor}\binom{r}{i}(s)^i(1-s)^{r-i} &\le \sum_{i=0}^{\lfloor r/2\rfloor}\binom{r}{i}(s)^{r/2}(1-s)^{r/2}\\ &\le (s-s^2)^{r/2}\sum_{i=0}^{\lfloor r/2\rfloor}\binom{r}{i}\\ &\le (s-s^2)^{r/2}.2^{-r}\\ &\le (4s-4s^2)^{r/2}\end{aligned}$$

We use the previous bound to choose r such that $(4s-4s^2)^{r/2} \le \delta$. In particular, for $s = 3/4$, we find:

$$r = \frac{2}{\log(4/3)}\log(1/\delta)$$

To recap: construct a single experiment Y_n using $n = O(\sigma_Y^2/\mu_Y^2\epsilon^{-2})$ samples, repeat the experiment $r = O(\log 1/\delta)$ times, and return median $\bar{\mu}_Y$. Using $O(\sigma_Y^2/\mu_Y^2\epsilon^{-2}\log 1/\delta)$ samples, we showed this procedure returns $\bar{\mu}_Y$ in the range $(1 \pm \epsilon)\cdot\mu_Y$ with at least probability $1-\delta$. □

The proof of Theorem 6 is adapted from Prof. Sinclair's online lecture notes [50].

References

[1] Ball MO. Computational complexity of network reliability analysis: an overview. IEEE Trans Reliab 1986;35(3):230–9.
[2] Valiant LG. The complexity of enumeration and reliability problems. SIAM J Comput 1979;8(3):410–21.
[3] Hardy G, Lucet C, Limnios N. K-Terminal network reliability measures with binary decision diagrams. IEEE Trans Reliab 2007;56(3):506–15.
[4] Canale E, Romero P, Rubino G. Factorization theory in diameter constrained reliability. 2016 8th international workshop on Resilient Networks Design and Modeling (RNDM). 6. IEEE978-1-4673-9023-1; 2016. p. 66–71.
[5] Fishman GS. A Monte Carlo sampling plan for estimating network reliability. Oper Res 1986;34(4):581–94.
[6] Valiant LG. A theory of the learnable. Commun ACM 1984;27(11):1134–42.
[7] Fishman GS. Monte Carlo: concepts, algorithms, and applications. New York, NY: Springer978-1-4419-2847-4; 1996.
[8] Gertsbakh IB, Shpungin Y. Models of network reliability: analysis, combinatorics, and Monte Carlo. CRC Press; 2016.
[9] Botev ZI, L'Ecuyer P, Rubino G, Simard R, Tuffin B. Static network reliability estimation via generalized splitting. INFORMS J Comput 2012;25(1):56–71.
[10] Zuev KM, Wu S, Beck JL. General network reliability problem and its efficient solution by subset simulation. Probab Eng Mech 2015;40:25–35.
[11] Cancela H, El Khadiri M. A recursive variance-reduction algorithm for estimating communication-network reliability. IEEE Trans Reliab 1995;44(4):595–602.
[12] Gertsbakh IB, Shpungin Y. Models ofnetwork reliability: analysis, combinatorics, and Monte Carlo. CRC Press; 2010.
[13] Vaisman R, Kroese DP, Gertsbakh IB. Splitting sequential Monte Carlo for efficient unreliability estimation of highly reliable networks. Struct Saf 2016;63:1–10.
[14] Karger DR. A Randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. SIAM Rev 2001;43(3):499–522.
[15] Meel KS. Constrained counting and sampling: bridging the gap between theory and practice. Rice University; 2017.
[16] Chakraborty S, Meel KS, Vardi MY. Algorithmic Improvements in Approximate Counting for Probabilistic Inference: From Linear to Logarithmic SAT Calls. Proceedings of the twenty-fifth international joint conference on artificial intelligence. New York, USA: AAAI Press978-1-57735-770-4; 2016. p. 3569–76.
[17] Stockmeyer L. The complexity of approximate counting. Proceedings of the fifteenth annual ACM symposium on theory of computing – STOC '83. New York, USA: ACM Press0897910990; 1983. p. 118–26.
[18] Bayer C, Hoel H, Von Schwerin E, Tempone R. On nonasymptotic optimal stopping criteria in Monte Carlo simulations. SIAM J Sci Comput 2014;36(2):A869–85.
[19] Ellingwood BR, van de Lindt JW, McAllister TP. Developing measurement science for community resilience assessment. Sustain Resil Infrastruct 2016;1(3–4):93–4.
[20] Duenas-Osorio L, Meel KS, Paredes R, Vardi MY. Counting-based reliability estimation for power-transmission grids. Proceedings of AAAI conference on artificial intelligence (AAAI). 2017. San Francisco
[21] Soos M, Meel KS. Bird: Engineering an efficient CNF-XOR sat solver and its applications to approximate model counting. Proceedings of AAAI conference on artificial intelligence (AAAI). 2019.
[22] Ball MO, Colbourn CJ, Provan JS. Chapter 11 network reliability. Handbooks in operations research and management science. 7. 19959780444892928. p. 673–762.
[23] Dotson W, Gobien J. A new analysis technique for probabilistic graphs. IEEE Trans Circt Syst 1979;26(10):855–65.
[24] Satyanarayana A, Chang MK. Network reliability and the factoring theorem. Networks 1983;13(1):107–20.
[25] Lê M, Weidendorfer J, Walter M. A novel variable ordering heuristic for BDD-based K-Terminal reliability. Proceedings – 44th annual IEEE/IFIP international conference on dependable systems and networks, DSN 2014. 2014. p. 527–37.
[26] Lê M, Walter M, Weidendorfer J. A memory-efficient bounding algorithm for the two-terminal reliability problem. Electron Notes Theor Comput Sci 2013;291:15–25.
[27] Lim H-W, Song J. Efficient risk assessment of lifeline networks under spatially correlated ground motions using selective recursive decomposition algorithm. Earthq Eng Struct Dyn 2012;41(13):1861–82.
[28] Paredes R, Dueñas-Osorio L, Hernandez-Fajardo I. Decomposition algorithms for system reliability estimation with applications to interdependent lifeline networks. Earthq Eng Struct Dyn 2018;47(13):2581–600.
[29] Alexopoulos C. A note on state-space decomposition methods for analyzing stochastic flow networks. IEEE Trans Reliab 1995;44(2):354–7.
[30] Cancela H, El Khadiri M, Rubino G, Tuffin B. Balanced and approximate zero-variance recursive estimators for the network reliability problem. ACM Trans Model ComputSimul 2014;25(1):1–19.
[31] Jerrum M, Sinclair A. Conductance and the rapid mixing property for Markov chains: the approximation of permanent resolved. Proceedings of the twentieth annual ACM symposium on Theory of computing – STOC '88. New York, USA: ACM Press0897912640; 1988. p. 235–44.
[32] Fishman GS. Markov chain sampling and the product estimator. Oper Res 1994;42(6):1137–45.
[33] Dyer M, Frieze A, Kannan R. A random polynomial-time algorithm for approximating the volume of convex bodies. J ACM 1991;38(1):1–17.
[34] Glasserman P, Heidelberger P. Multilevel splitting for estimating rare event probabilities. Oper Res 1999;47(4):585–600.
[35] Kahn H, Harris TE. Estimation of particle transmission by random sampling. National Bureau of Standards applied mathematics series 12. 1951. p. 27–30.
[36] Rosenbluth MN, Rosenbluth AW. Monte Carlo calculation of the average extension of molecular chains. J Chem Phys 1955;23(2):356–9.
[37] Au SK, Beck JL. Estimation of small failure probabilities in high dimensions by subset simulation. Probab Eng Mech 2001;16(4):263–77.
[38] Karger DR. A fast and simple unbiased estimator for network (Un)reliability. 2016 IEEE 57th annual symposium on foundations of computer science (FOCS). 2016. p. 635–44.
[39] Dagum P, Karp R, Luby M, Ross S. An optimal algorithm for Monte Carlo estimation. Proceedings of IEEE 36th annual foundations of computer science. 29. 1995. p. 1–22.
[40] Huber M. A bernoulli mean estimate with known relative error distribution. Random Struct Algorithms 2017;50(2):173–82.
[41] Wald A. Sequential analysis. New York: John Wiley & Sons; 1947.
[42] Herrmann JU. Improving reliability calculation with augmented binary decision diagrams. 2010 24th IEEE international conference on advanced information networking and applications. IEEE978-1-4244-6695-5; 2010. p. 328–33.
[43] Fishman GS. A comparison of four Monte Carlo methods for estimating the probability of s-t connectedness. IEEE Trans Reliab 1986;35(2):145–55.
[44] Dueñas-Osorio L, Vardi M, Rojo J. Quantum-inspired boolean states for bounding engineering network reliability assessment. Struct Saf 2018;75(April 2017):110–8.
[45] Li J, Dueñas-Osorio L, Chen C, Berryhill B, Yazdani A. Characterizing the topological and controllability features of U.S. power transmission networks. Physica A 2016;453:84–98.
[46] Gertsbakh I, Neuman E, Vaisman R. Monte Carlo for estimating exponential convolution. Commun Stat– SimulComput 2015;44(10):2696–704.
[47] Chan TF, Golub GH, LeVeque RJ. Algorithms for computing the sample variance: analysis and recommendations. Am Stat 1983;37(3):242.
[48] Karger DR, Stein C. A new approach to the minimum cut problem. J ACM 1996;43(4):601–40.
[49] Preskill J. Quantum computing in thek NISQ era and beyond. Quantum 2018;2:79.
[50] Sinclair A. Lecture 11: counting satisfying assignments of a DNF formula. Berkeley: University of California; 2018. (CS 271)