

Efficient estimation of complex network reliability

K. M. Zuev

University of Liverpool, UK

S. Wu & J. L. Beck

California Institute of Technology, USA

ABSTRACT: Complex technological networks designed for distribution of some resource or commodity are a pervasive feature of modern society. Moreover, the dependence of our society on modern technological networks constantly grows. As a result, there is an increasing demand for these networks to be highly reliable. The degree to which a network is able to provide the required service needs to be quantitatively estimated. This quantitative assessment of network reliability is a very important aspect of network design, optimization, and operation. In this paper, we propose a framework for quantitative assessment of network reliability, formulate a general network reliability problem within this framework, and propose an efficient Markov chain Monte Carlo method for solving this problem, which is based on Subset Simulation, a very efficient stochastic simulation method originally developed for estimation of small failure probabilities of complex civil engineering structures such as tall buildings and bridges at risk from earthquakes. The efficiency of the method is demonstrated with an illustrative example where two small-world network models are compared in terms of reliability of networks they produce.

1 INTRODUCTION

Complex technological networks are a pervasive feature of modern society. The worldwide increase in urbanization and globalization, accompanied by rapid growth of infrastructure and technology, has produced complex networks with ever more interdependent components. These networks are designed for distribution of some resource or commodity. Examples include transportation networks (e.g. networks of roads or rails, or networks of airline routes), communication networks (e.g. telephone networks or the Internet), utility networks (e.g. networks for delivery of electricity, gas or water), etc.

Systems in the form of technological networks are an integral part of megacities; their failures, although rare, have serious consequences on the wellbeing of society. The dependence of our society on technological systems and information networks constantly grows. Nowadays we can hardly imagine our life without electricity, telephones, airplanes, and even without the Internet. As a result, there is an increasing demand for modern technological networks to be highly reliable. The degree to which a network is able to provide the required service needs to be quantita-

tively estimated. This quantitative assessment of network reliability is a very important aspect of network design, optimization, and operation.

Traditional methods for network reliability analyses are based on graph theory and mostly look at small scale networks. These methods aim to exactly compute the network reliability and can be roughly classified by the following (not mutually exclusive) three categories: enumeration methods, direct methods, and decomposition methods. Enumeration methods are typically based on either complete state enumeration or (more sophisticated) a minpath or mincut enumeration. Direct methods are intended to compute the reliability of a network from the structure of the underlying graph, without a preliminary search for the minpaths and mincuts. In decomposition methods, the main idea is to divide the network into several subnetworks, and the overall reliability is then calculated based on the reliabilities of the corresponding subnetworks. A detailed review of traditional methods for reliability analysis of small scale network is provided in (Jonczy & Haenni 2007). All these methods in one way or another are based on combinatorial exhaustive search through the network.

On the other hand, one of the inherent characteristic features of modern technological networks is their very large size. Today the complexity of real-world networks can reach millions or even billions of vertices and edges with incomprehensible topology.

This dramatic change of scale induced a corresponding change in the philosophy of reliability analyses. Many of the exhaustive search algorithms that have been applied to small networks are simply not feasible for large networks, since essentially all reliability problems of interest are NP-hard (Rosenthal 1977) and the exhaustive algorithms grow in complexity very rapidly as a function of the network size. It has been thus recognized that the classical methods of reliability and risk analysis fail to provide the proper instruments for analysis of actual modern networks (Zio 2007). As a result, a new field of research has recently emerged with the focus shifting away from the combinatorial exhaustive search methodology to the study of statistical properties of large networks, together with the study of their robustness to random failures, errors, and intentional attacks.

In this paper, we propose a framework for quantitative assessment of network reliability, formulate a general network reliability problem within this framework, and propose an efficient Markov chain Monte Carlo method for solving this problem, which is based on Subset Simulation (Au & Beck 2001), a very efficient stochastic simulation method originally developed for estimation of small failure probabilities of complex civil engineering structures such as tall buildings and bridges at risk from earthquakes.

We proceed as follows. In the next section, we formulate a general network reliability problem subjected to several realistic conditions that make this problem computationally difficult. In Section 3, we describe the Subset Simulation algorithm for solving the network reliability problem. An illustrative example that demonstrates how Subset Simulation can be effectively used for solving the maximum-flow reliability problem and for finding reliable network topologies is provided in Section 4. Concluding remarks are made in Section 5.

2 NETWORK RELIABILITY PROBLEM

A network topology is represented as a graph $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$ are sets of n nodes (or vertices) and m links (or edges), respectively. Any graph G with n nodes can be represented by its $n \times n$ adjacency matrix $A(G)$, where $A_{ij} = 1$ if there is a link connecting v_i to v_j ($i \neq j$) and $A_{ij} = 0$ otherwise. The degree of a node $v_i \in V$, denoted $d(v_i)$, is the number of links incident to the node and it is equal to the i^{th} row sum of A . The graph G may be either *undirected* or *directed*. In the latter case, the adjacency matrix $A(G)$ is not necessarily symmetric. In this case, node degree $d(v)$ is replaced by *in-degree* $d_{in}(v)$ and *out-degree*

$d_{out}(v)$, which count the number of links pointing in towards and out from a node, respectively. In general, any undirected network can be considered as directed after replacement of every undirected link by two corresponding opposing directed links.

In what follows we assume that nodes are perfectly reliable, i.e. they have zero failure probability. It can be shown that any network with node failures is polynomial-time reducible into an equivalent directed network with link failures only (Colbourn 1987). Thus, this assumption does not, in fact, limit the generality, and networks with link failures only are sufficiently general.

A *network state* is defined as an m -tuple $s = (s_1, \dots, s_m)$ with $s_i \in [0, 1]$, where $s_i = 1$ if link e_i is fully operational (or “up”) and $s_i = 0$ if link e_i is completely failed (or “down”). If $s_i \in (0, 1)$, then link e_i is partially operational. The set of all network states \mathcal{S} is then an m -dimensional hypercube,

$$\mathcal{S} = \{(s_1, \dots, s_m) \mid s_i \in [0, 1]\} = [0, 1]^m \quad (1)$$

Let $\pi(s)$ be a probability distribution on the network state space \mathcal{S} which provides a probability model for the occurrence of different network states, $s \sim \pi(s)$. For example, if we assume that each link e_i is either up ($s_i = 1$) or down ($s_i = 0$) and that links fail independently of each other with failure probabilities p_i , then this induces a discrete probability distribution on \mathcal{S} : each state $s = (s_1, \dots, s_m)$ with $s_i = 0, 1$ has occurrence probability

$$\pi(s) = \prod_{i=1}^m q_i^{s_i} p_i^{1-s_i}, \quad (2)$$

where $q_i = 1 - p_i$ is the edge reliability; partially operational links do not occur in this model. In general case, $\pi(s)$ is a continuous probability model on \mathcal{S} , and its choice depends on a particular network.

Furthermore, we define a *performance function* $\mu : \mathcal{S} \rightarrow \mathbb{R}$ that quantifies the degree to which the network provides the required service. In the context of networks, μ is typically interpreted as a utility function, i.e. higher values of μ correspond to better network performance. Let us define the *failure domain* $\mathcal{F} \subset \mathcal{S}$ as follows:

$$\mathcal{F} = \{s \in \mathcal{S} \mid \mu(s) < \mu^*\}, \quad (3)$$

where μ^* is the critical threshold.

The *network reliability problem* is to compute the probability of failure $p_{\mathcal{F}}$, that is given by the following integral:

$$p_{\mathcal{F}} = \mathbb{P}(s \in \mathcal{F}) = \int_{\mathcal{S}} \pi(s) I_{\mathcal{F}}(s) ds = \mathbb{E}_{\pi}[I_{\mathcal{F}}]. \quad (4)$$

Several classical reliability problems (Ball, Colbourn, & Provan 1995, Rosenthal 1977, Colbourn

1987) are special cases of the above general formulation, e.g. Source-to-Terminal Connectedness, Network Connectedness, Traffic to Central Site, to name but a few.

We make the following real-life assumptions:

- (i) *The computational effort for evaluating the network performance function $\mu(s)$ for each state $s \in \mathcal{S}$ is significant*, thereby making the indicator function $I_{\mathcal{F}}(s)$ expensive to compute. Therefore, it is essential to minimize the number of such function evaluations.
- (ii) *The number of edges m is large, i.e. $m \gg 1$* . Many actual networks have millions (e.g. road networks), or even billions, of edges (e.g. the Internet).
- (iii) *The probability of failure $p_{\mathcal{F}}$ is very small, i.e. $p_{\mathcal{F}} \ll 1$* . Real-life networks are reliable to some extent (otherwise they would not be in use), and their failures are usually rare events.

These assumptions make the network reliability problem computationally very challenging. Due to (i) and (ii), the integral in (4) is taken over a high-dimensional hypercube and the integrand is expensive to compute. Therefore, the exact computation of the failure probability $p_{\mathcal{F}}$ is infeasible.

The expression of $p_{\mathcal{F}}$ as a mathematical expectation (4) renders standard Monte Carlo method directly applicable, where $p_{\mathcal{F}}$ is estimated as a sample average of $I_{\mathcal{F}}(s)$ over independent and identically distributed samples of s drawn from distribution $\pi(s)$:

$$\hat{p}_{\mathcal{F}}^{(MC)} = \frac{1}{N} \sum_{i=1}^N I_{\mathcal{F}}(s^{(i)}), \quad s^{(i)} \sim \pi(s) \quad (5)$$

This estimate is unbiased and the coefficient of variation, serving as a measure of the statistical error, is

$$\delta_{MC} = \sqrt{\frac{1-p_{\mathcal{F}}}{N p_{\mathcal{F}}}} \quad (6)$$

Although standard Monte Carlo is independent of the dimension m of the network state space \mathcal{S} , it is inefficient in estimating small probabilities because it requires a large number of samples ($\sim 1/p_{\mathcal{F}}$) to achieve an acceptable level of accuracy. For example, if $p_{\mathcal{F}} = 10^{-4}$ and we want to achieve an accuracy of $\delta_{MC} = 10\%$, then we need approximately $N = 10^6$ samples. Therefore, due to conditions (i) and (iii), standard Monte Carlo becomes computationally prohibitive for our problems of interest involving small failure probabilities.

3 SUBSET SIMULATION

The main idea of Subset Simulation (Au & Beck 2001) is to represent a small failure probability $p_{\mathcal{F}}$ as a product $p_{\mathcal{F}} = \prod_{j=1}^L p_j$ of larger probabilities $p_j > p_{\mathcal{F}}$, where the factors p_j are estimated sequentially, $p_j \approx \hat{p}_j$ to obtain an estimate $\hat{p}_{\mathcal{F}}$ for $p_{\mathcal{F}}$ as $\hat{p}_{\mathcal{F}} = \prod_{j=1}^L \hat{p}_j$. To achieve this goal, let us consider a sequence of nested subsets of the network state space \mathcal{S} , starting from the entire space and shrinking to the failure domain:

$$\mathcal{S} = \mathcal{F}_0 \supset \mathcal{F}_1 \supset \dots \supset \mathcal{F}_L = \mathcal{F} \quad (7)$$

Subsets $\mathcal{F}_0, \dots, \mathcal{F}_{L-1}$ are called *intermediate failure domains*. The failure probability can be written then as a product of conditional probabilities:

$$p_{\mathcal{F}} = \prod_{j=1}^L \mathbb{P}(\mathcal{F}_j | \mathcal{F}_{j-1}) = \prod_{j=1}^L p_j, \quad (8)$$

where $p_j = \mathbb{P}(\mathcal{F}_j | \mathcal{F}_{j-1})$ is the conditional probability at the $(j-1)^{\text{th}}$ conditional level. Clearly, by choosing the intermediate failure domains $\mathcal{F}_1, \dots, \mathcal{F}_{L-1}$ appropriately, all conditional probabilities p_1, \dots, p_L can be made sufficiently large. The original network reliability problem (estimation of the small failure probability $p_{\mathcal{F}}$) is thus replaced by a sequence of L intermediate problems: estimation of the larger failure probabilities p_j , $j = 1, \dots, L$.

The first probability $p_1 = \mathbb{P}(\mathcal{F}_1 | \mathcal{S}) = \mathbb{P}(\mathcal{F}_1)$ can be simply estimated by standard Monte Carlo simulation (MCS):

$$p_1 \approx \hat{p}_1 = \frac{1}{N} \sum_{i=1}^N I_{\mathcal{F}_1}(s_0^{(i)}), \quad (9)$$

$$s_0^{(i)} \stackrel{i.i.d.}{\sim} \pi(s | \mathcal{F}_0) \equiv \pi(s)$$

We assume here that \mathcal{F}_1 is chosen in such a way that p_1 is relatively large, so that the MCS estimate (9) is accurate for a moderate sample size N . Later in this section, we will discuss how to choose intermediate failure domains \mathcal{F}_j adaptively.

For $j \geq 2$, to estimate p_j using MSC one needs to simulate i.i.d. samples from conditional distribution $\pi(s | \mathcal{F}_{j-1})$, which, for general $\pi(s)$ and \mathcal{F}_{j-1} , is not a trivial task. For example, it would be inefficient to use MCS for this purpose (i.e. to sample from $\pi(s)$ and accept only those samples that belong to \mathcal{F}_{j-1}), especially at higher levels. Sampling from $\pi(s | \mathcal{F}_{j-1})$ for $j \geq 2$ can be done by a specifically tailored MCMC technique at the expense of generating dependent samples.

The Metropolis-Hastings (MH) algorithm (Hastings 1970) is perhaps the most popular MCMC algorithm for sampling from a probability distribution that is difficult to sample from directly. In this algorithm

samples are generated as the states of a Markov chain, which has the target distribution, i.e. the distribution we want to sample from, as its stationary distribution. In our case, the target distribution is $\pi(s|\mathcal{F}_{j-1}) = \pi(s)I_{\mathcal{F}_{j-1}}(s)/\mathcal{Z}_{j-1}$, where $\mathcal{Z}_{j-1} = \mathbb{P}(\mathcal{F}_{j-1})$ is a normalizing constant. Let $s_{j-1}^{(i)}$ be the current state of the Markov chain and $q(s|s_{j-1}^{(i)})$, called the proposal PDF, be an m -dimensional PDF on the network state space \mathcal{S} that depends on $s_{j-1}^{(i)}$ and can be readily sampled. Then the MH update $s_{j-1}^{(i)} \rightarrow s_{j-1}^{(i+1)}$ of the Markov chain works as follows:

1. Generate a candidate state $\xi \in \mathcal{S}$ according to $q(s|s_{j-1}^{(i)})$,
2. Compute the acceptance probability $\alpha = I_{\mathcal{F}_{j-1}}(\xi) \min \left\{ 1, \frac{\pi(\xi)q(s_{j-1}^{(i)}|\xi)}{\pi(s_{j-1}^{(i)})q(\xi|s_{j-1}^{(i)})} \right\}$
3. Accept ξ as the next state of the Markov chain, i.e. set $s_{j-1}^{(i+1)} = \xi$, with probability α , or reject it, i.e. set $s_{j-1}^{(i+1)} = s_{j-1}^{(i)}$ with the remaining probability $1 - \alpha$.

It is easy to show that this update leaves $\pi(s|\mathcal{F}_{j-1})$ invariant, i.e. if $s_{j-1}^{(i)}$ is distributed according to $\pi(s|\mathcal{F}_{j-1})$, then so is $s_{j-1}^{(i+1)}$, and if the Markov chain is run for sufficiently long time (the "burn-in period"), starting from essentially any "seed" $s_{j-1}^{(1)} \in \mathcal{F}_{j-1}$, then for large N the distribution of $s_{j-1}^{(N)}$ will be approximately $\pi(s|\mathcal{F}_{j-1})$. Note, however, that usually it is very difficult to check whether the Markov chain has converged to its stationary distribution. But if the seed $s_{j-1}^{(1)} \sim \pi(s|\mathcal{F}_{j-1})$, then all states of the Markov chain will be automatically distributed according to the target distribution, $s_{j-1}^{(i)} \sim \pi(s|\mathcal{F}_{j-1}), i = 1, \dots, N$. The absence of the burn-in period (i.e. the absence of the convergence problem) is often referred to as *perfect sampling* (Robert & Casella 2004) and Subset Simulation has this property because of the way the seeds are selected.

It was observed in (Au & Beck 2001) however, that the original Metropolis-Hastings algorithm suffers from the curse of dimensionality. Namely, it is not efficient in high-dimensional conditional probability spaces, because it produces a Markov chain with very highly correlated states. Therefore, if the total number of network links m is large, then the MH algorithm will be inefficient for sampling from $\pi(s|\mathcal{F}_{j-1})$, where $\mathcal{F}_{j-1} \subset \mathcal{S} = [0, 1]^m$. In Subset Simulation, the Modified Metropolis algorithm (MMA) (Au & Beck 2001), an MCMC technique based on the original MH algorithm is used instead for sampling from the conditional distributions $\pi(s|\mathcal{F}_{j-1})$. MMA differs from the MH algorithm in the way the candidate state $\xi = (\xi_1, \dots, \xi_m)$ is generated. Instead of using an m -dimensional proposal PDF on \mathcal{S} to directly

obtain the candidate state, in MMA a sequence of univariate proposal PDFs is used. Namely, each coordinate ξ_k of the candidate state is generated separately using a univariate proposal distribution $q_k(s_k|s_{j-1,k}^{(i)})$ dependent on the k^{th} coordinate $s_{j-1,k}^{(i)}$ of the current state. Then a check is made whether the m -variate candidate $\xi \in \mathcal{S}$ generated in such a way belongs to the subset \mathcal{F}_{j-1} in which case it is accepted as the next Markov chain state; otherwise it is rejected and the current MCMC sample is repeated. For details on MMA, we refer the reader to the original paper (Au & Beck 2001) and to (Zuev, Beck, Au, & Katafygiotis 2012) where the algorithm is discussed in depth.

Let us assume now that we are given a seed $s_{j-1}^{(1)} \sim \pi(s|\mathcal{F}_{j-1})$, where $j = 2, \dots, L$. Then, using MMA, we can generate a Markov chain with N states starting from this seed and construct an estimate for p_j similar to (9), where MCS samples are replaced by MCMC samples:

$$p_j \approx \hat{p}_j = \frac{1}{N} \sum_{i=1}^N I_{\mathcal{F}_j}(s_{j-1}^{(i)}), \quad (10)$$

$$s_{j-1}^{(i)} \stackrel{\text{MMA}}{\sim} \pi(s|\mathcal{F}_{j-1})$$

Note that all samples $s_{j-1}^{(1)}, \dots, s_{j-1}^{(N)}$ in (10) are identically distributed in the stationary state of the Markov chain, but are not independent. Nevertheless, these MCMC samples can be used for statistical averaging as if they were i.i.d., although with some reduction in efficiency (Doob 1953).

Finally, Subset Simulation uses the estimates (9) for p_1 and (10) for $p_j, j \geq 2$, to obtain the estimate for the failure probability:

$$p_{\mathcal{F}} \approx \hat{p}_{\mathcal{F}} = \prod_{j=1}^L \hat{p}_j \quad (11)$$

The remaining ingredient of Subset Simulation that we have to discuss is the choice of intermediate failure domains $\mathcal{F}_1, \dots, \mathcal{F}_{L-1}$. Recall that the "ultimate" failure domain $\mathcal{F} = \mathcal{F}_L \in \mathcal{S}$ is defined as $\mathcal{F} = \{s \in \mathcal{S} \mid \mu(s) < \mu^*\}$. The sequence of intermediate failure domains can then be defined in a similar way:

$$\mathcal{F}_j = \{s \in \mathcal{S} \mid \mu(s) < \mu_j^*\}, \quad (12)$$

where $\mu^* = \mu_L^* < \mu_{L-1}^* < \dots < \mu_1^*$ is a sequence of intermediate critical thresholds. Intermediate threshold values μ_j^* define the values of the conditional probabilities $p_j = \mathbb{P}(\mathcal{F}_j|\mathcal{F}_{j-1})$ and, therefore, affect the efficiency of Subset Simulation. In practical cases it is difficult to make a rational choice of the μ_j^* -values in advance, so the μ_j^* are chosen adaptively (see (13) below) so that the estimated conditional probabilities are equal to a fixed value $p_0 \in (0, 1)$. In (Zuev, Beck,

Au, & Katafygiotis 2012), p_0 is called the conditional failure probability.

SubSim for Network Reliability Problem

Input:

- ▷ p_0 , conditional failure probability;
- ▷ N , number of samples per conditional level.

Algorithm:

Set $j = 0$, number of conditional level.

Set $N_{\mathcal{F}}(j) = 0$, number of failure samples at level j .

Sample $s_0^{(1)}, \dots, s_0^{(N)} \stackrel{i.i.d.}{\sim} \pi(s)$.

for $i = 1, \dots, N$ **do**

if $\mu^{(i)} = \mu(s_0^{(i)}) < \mu^*$ **do**

$N_{\mathcal{F}}(j) \leftarrow N_{\mathcal{F}}(j) + 1$

end if

end for

while $N_{\mathcal{F}}(j)/N < p_0$ **do**

$j \leftarrow j + 1$

 Sort $\{\mu^{(i)}\}$: $\mu^{(i_1)} \leq \mu^{(i_2)} \leq \dots \leq \mu^{(i_N)}$

 Define

$$\mu_j^* = \frac{\mu^{(i_{Np_0})} + \mu^{(i_{Np_0+1})}}{2} \quad (13)$$

for $k = 1, \dots, Np_0$ **do**

 Starting from $s_j^{(1),k} = s_{j-1}^{(i_k)} \sim \pi(s|\mathcal{F}_j)$, generate $1/p_0$ states of a Markov chain

$s_j^{(1),k}, \dots, s_j^{(1/p_0),k} \sim \pi(s|\mathcal{F}_j)$, using MMA.

end for

 Renumber:

$\{s_j^{(i),k}\}_{k=1, i=1}^{Np_0, 1/p_0} \mapsto s_j^{(1)}, \dots, s_j^{(N)} \sim \pi(s|\mathcal{F}_j)$

for $i = 1, \dots, N$ **do**

if $\mu^{(i)} = \mu(s_j^{(i)}) < \mu^*$ **do**

$N_{\mathcal{F}}(j) \leftarrow N_{\mathcal{F}}(j) + 1$

end if

end for

end while

Output:

- ▶ $\hat{p}_{\mathcal{F}}$, estimate of $p_{\mathcal{F}}$:

$$\hat{p}_{\mathcal{F}} = p_0 \frac{N_{\mathcal{F}}(j)}{N} \quad (14)$$

The adaptive choice of μ_j^* -values in (13) guarantees, first, that all seeds $s_j^{(1),k}$ are distributed according to $\pi(s|\mathcal{F}_j)$ and, second, that the estimated conditional probability $\mathbb{P}(\mathcal{F}_j|\mathcal{F}_{j-1})$ is equal to p_0 . Here, for convenience, p_0 is assumed to be chosen such that Np_0 and $1/p_0$ are positive integers, although this is not strictly necessary. It was demonstrated in (Zuev, Beck, Au, & Katafygiotis 2012) that choosing any $p_0 \in [0.1, 0.3]$ will lead to similar efficiency and it is not necessary to fine tune the value of the conditional failure probability as long as Subset Simulation is implemented properly.

4 EXAMPLE: MAXIMUM FLOWS IN SMALL-WORLD NETWORKS

In this section we illustrate how Subset Simulation can be used for solving the maximum-flow reliability problem and for finding reliable network topologies.

4.1 The maximum-flow reliability problem

Network flow problems naturally arise in many real world applications such as coordination of vehicles in a transportation network, distribution of water in a utility network, and routing of packets in a communication network. The maximum-flow problem, where the goal is to maximize the total flow from one node of a network to another, is one of the classical network flow problems (Ford & Fulkerson 1962).

Suppose that in addition to a network $G = (V, E)$, a distinguished pair of nodes, the *source* $a \in V$ and the *sink* $b \in V$, is specified. Also assume that each link $e = (v, u) \in E$ has a non-negative *capacity* $c(v, u) \geq 0$. A quadruple $(G, a, b, \{c\})$ is often referred to as a *flow network*. A *flow* on G is non-negative function $f: E \rightarrow \mathbb{R}^+$ that satisfies the following properties:

- *Capacity constraint*: the flow along any link cannot exceed the capacity of that link,

$$f(v, u) \leq c(v, u), \quad \text{for all } (v, u) \in E \quad (15)$$

- *Flow conservation*: the total flow entering node v must equal the total flow leaving v for all nodes except a and b ,

$$\sum_{u \in V} f(u, v) = \sum_{u \in V} f(v, u), \quad v \in V \setminus \{a, b\} \quad (16)$$

For convenience, it is assumed in (16), that $f(v, u) = 0$ if $(v, u) \notin E$.

The *value* $|f|$ of a flow f is the net flow out of the source:

$$|f| = \sum_{v \in V} f(a, v) - \sum_{v \in V} f(v, a) \quad (17)$$

It is easy to show that $|f|$ also equals the net flow into the sink,

$$|f| = \sum_{v \in V} f(v, b) - \sum_{v \in V} f(b, v) \quad (18)$$

The value of a flow represents how much one can transport from the source to the sink.

The *maximum-flow problem* is that of finding the maximum flow $f^* = \arg \max |f|$ in a given flow network $(G, a, b, \{c\})$. There is a simple algorithm called the augmented path algorithm (also called the Ford-Fulkerson algorithm) that calculates the maximum

flows between nodes in polynomial time. The theory of maximum flow algorithms is well covered in (Ahuja, Magnati, & Orlin 1962).

The *maximum-flow reliability problem* considered in this paper is motivated by the maximum-flow problem. Assume for convenience that all link capacities s_1, \dots, s_m are normalized, $0 \leq s_i \leq 1$, and suppose that, instead of being prescribed, they are uncertain. For example, due to environmental degradation, pipes in a water distribution network may leak at an unknown rate. Let π be a probability model for link capacities, i.e. a probability distribution on the network state space $\mathcal{S} = \{s = (s_1, \dots, s_m) \mid 0 \leq s_i \leq 1\}$. For a given realization $s \sim \pi(s)$, we define the max-flow network performance function to be equal to the value of the corresponding maximum flow:

$$\mu_{MF}(s) = |f^*(s)| \quad (19)$$

The failure domain $\mathcal{F} \subset \mathcal{S}$ can now be defined as before, $\mathcal{F} = \{s \in \mathcal{S} \mid \mu_{MF}(s) < \mu^*\}$, where μ^* is the critical threshold. Thus, the introduction of uncertain link capacities brings us into the network reliability framework described in (4): $p_{\mathcal{F}} = \mathbb{P}(s \in \mathcal{F})$.

One of the fundamental questions in network science and reliability theory is the following: given n nodes and m links, how to combine them into the most reliable network? In the next section we analyze a simpler but related question: given two network models that generate componentwise equivalent but topologically different networks, how to find out what network model produces a more reliable network?

4.2 Small-world network models

One of the most important breakthroughs in modeling real-world networks was a shift from classical random graph models, where links between nodes are placed purely at random, to models that explicitly mimic certain statistical properties observed in actual networks. Small-world network models were originally inspired by the counter intuitive phenomenon observed by the social psychologist Stanley Milgram in human social networks (Milgram 1967). In his famous experiment, each of the participants (randomly chosen in Nebraska) was asked to forward a letter to one of their friends in an attempt to get the letter ultimately to a desired target person (who lived in Boston, Massachusetts). The obtained results were very surprising: the average number of people needed to transmit the letter to the target was approximately six. This gave birth to the popular belief that there are only about six handshakes between any two people in the world, so-called “six degrees of separation.” Milgram’s experiment was one of the first quantitative observations of the *small-world effect*, the fact that despite their often large size and high level of clustering, in most actual networks there is a relatively short path between almost any two nodes. The small-world effect has been observed in many real

networks, including technological ones such as power grids (Watts & Strogatz 1998), airline routes (Amaral, Scala, Barthelemy, & Stanley 2000), and railways (Sen, Dasgupta, Chatterjee, Sreeram, Mukherjee, & Manna 2003).

In their seminal paper, (Watts & Strogatz 1998) proposed the first network model (the WS model) that generates “small-worlds”, i.e. networks with small average shortest-path lengths and high levels of clustering. The original WS model is a one-parameter model which interpolates between a regular lattice and a random graph. The model starts from a one-dimensional lattice of n nodes with periodic boundary conditions, i.e. a ring lattice. Each node is then connected to its first $2k$ neighbors (k on either side), with $k \ll n$. Thus we obtain a regular symmetric lattice with $m = nk$ links. The small-world network is then created by taking each link and turn and, with probability p , rewiring one end of that link to a new node chosen uniformly at random. The rewiring process generates on average pnk long-range connections. Note that, as p goes from 0 to 1, the model moves from a deterministic regular lattice to a random graph. For $0 < p < 1$, the WS model generates networks with the small-world property.

Since the pioneering work of Watts and Strogatz, many modifications of the WS model have been proposed and studied. In this paper, we consider two small-world network models that are built on one- and two-dimensional lattices with periodic boundary conditions, the small-world ring model and the small-world torus model, respectively.

- **Small-World Ring model** $\otimes(n, k)$

As the WS model, $\otimes(n, k)$ starts with a ring lattice of n nodes. This lattice has n undirected links, which equivalently can be considered as $2n$ directed links. Next, for each node v , the model generates k additional directed links connecting that node with other k nodes v_1, \dots, v_k chosen uniformly at random. Note that for each selected v_i , either link (v, v_i) or (v_i, v) is placed with equal probabilities.

- **Small-World Torus model** $\boxtimes(n, k)$

This model starts with a square $n \times n$ lattice. Periodic boundary conditions make it topologically equivalent to a 2-D torus, hence the name of the model. This torus has $2n^2$ undirected links, or, equivalently, $4n^2$ directed links. The small-world torus $\boxtimes(n, k)$ is then created by adding k directed links per each node in the same way as in the case of $\otimes(n, k)$ above.

Hereafter, we refer to deterministic lattice links as the *regular links* and to the randomly generated links as the *random shortcuts*. Realizations of $\otimes(16, 3)$ and $\boxtimes(4, 1)$ are schematically shown in Figures 1(a) and 1(b), respectively.

Figure 1: (a) Realization of the small-world ring model $\otimes(n, k)$, with $n = 16$ and $k = 3$. Solid and dashed lines represent regular links and random shortcuts, respectively. For visibility, only random shortcuts that correspond to nodes 1, 5, 9, and 13 are shown; (b) Realization of the small-world torus model $\boxtimes(n, k)$, with $n = 4$ and $k = 1$. Solid and dashed lines represent regular links and random shortcuts, respectively. For visibility, only random shortcuts that correspond to nodes 1, 4, 13, and 16 are shown.

It is easy to see that $\otimes(n_1, k_1)$ and $\boxtimes(n_2, k_2)$ produce componentwise equivalent networks if and only if $n_1 = n_2^2$ and $k_1 = k_2 + 2$. Topologically, however, the network realizations of these models will still be different, since the underlying lattices have different dimensions. Thus, we have:

$$\text{Componentwise: } \otimes(n^2, k+2) = \boxtimes(n, k) \quad (20)$$

$$\text{Topologically: } \otimes(n^2, k+2) \neq \boxtimes(n, k)$$

The small-world torus model $\boxtimes(n, k)$ has more regular links, while the small-world ring model $\otimes(n^2, k+2)$ has more random shortcuts. This split between the number of regular links versus random shortcuts, that represents the tradeoff between local connectivity and global reachability, has the potential to yield significantly different reliability properties for the two network models. To demonstrate the effectiveness of Subset Simulation for network reliability estimation, we investigate the following question: what model, $\otimes(n^2, k+2)$ or $\boxtimes(n, k)$, produces a more reliable network, where reliability is understood as the maximum-flow reliability?

4.3 Simulation results

Reliabilities of two network models with different topologies can be compared in the following way. Given a network realization $\hat{\otimes} \sim \otimes(n^2, k+2)$, a source-sink pair (a, b) , and the critical threshold μ^* , we can estimate the network failure probability $p_{\mathcal{F}}(\hat{\otimes}; (a, b); \mu^*)$ with respect to the max-flow performance function (19), using Subset Simulation described in Section 3. By averaging over different network realizations and source-sink pairs, we obtain the expected failure probability for a given critical threshold for the small-world ring model:

$$\begin{aligned} \bar{p}_{\mathcal{F}, \otimes}(\mu^*) &= \mathbb{E}_{\otimes, (a, b)} [p_{\mathcal{F}}(\hat{\otimes}; (a, b); \mu^*)] \\ &\approx \frac{1}{M} \sum_{i=1}^M p_{\mathcal{F}}(\hat{\otimes}_i; (a_i, b_i); \mu^*) \end{aligned} \quad (21)$$

Similarly we can estimate the expected failure probability for the small-world torus model:

$$\begin{aligned} \bar{p}_{\mathcal{F}, \boxtimes}(\mu^*) &= \mathbb{E}_{\boxtimes, (a, b)} [p_{\mathcal{F}}(\hat{\boxtimes}; (a, b); \mu^*)] \\ &\approx \frac{1}{M} \sum_{i=1}^M p_{\mathcal{F}}(\hat{\boxtimes}_i; (a_i, b_i); \mu^*), \end{aligned} \quad (22)$$

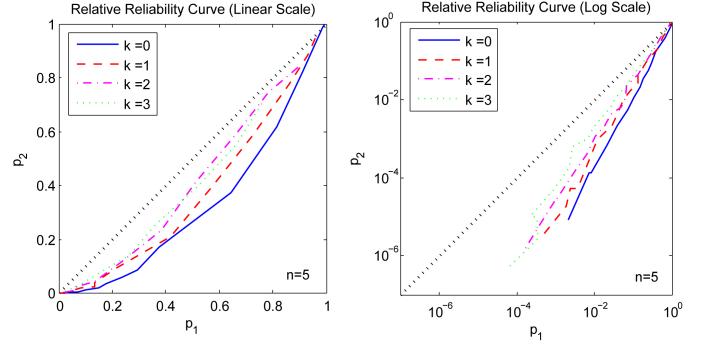


Figure 2: The relative reliability curve in the linear scale (left panel) and log scale (right panel), for $n=5$. Here, $p_1 = \bar{p}_{\mathcal{F}, \otimes}$ and $p_2 = \bar{p}_{\mathcal{F}, \boxtimes}$ as in (21) and (22) for $\mu^* \in (0.1, 3)$.

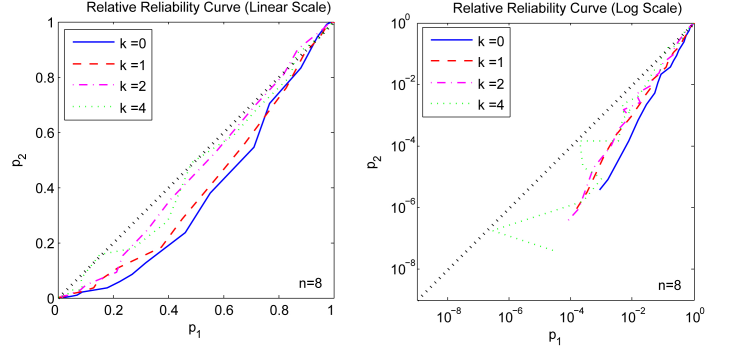


Figure 3: The relative reliability curve in the linear scale (left panel) and log scale (right panel), for $n=8$. Here, $p_1 = \bar{p}_{\mathcal{F}, \otimes}$ and $p_2 = \bar{p}_{\mathcal{F}, \boxtimes}$ as in (21) and (22) for $\mu^* \in (0.1, 6)$.

where $\hat{\boxtimes}_i \sim \boxtimes(n, k)$ are i.i.d, and node pairs (a_i, b_i) are chosen uniformly at random.

Since our goal is to compare reliabilities of networks that are produced by the two network models, we are interested not in the functions $\bar{p}_{\mathcal{F}, \otimes}(\mu^*)$ and $\bar{p}_{\mathcal{F}, \boxtimes}(\mu^*)$ *per se*, but rather in their relative behavior. We can achieve this by treating the critical threshold μ^* as a parameter and plotting $\bar{p}_{\mathcal{F}, \otimes}$ versus $\bar{p}_{\mathcal{F}, \boxtimes}$. The resulting curve will lie in the unit square, since both probabilities are between 0 and 1; it starts at $(0, 0)$, since both probabilities converge to 0, as $\mu^* \rightarrow -\infty$; and it ends at $(1, 1)$, since both probabilities converge to 1, as $\mu^* \rightarrow +\infty$. We refer to this curve as the *relative reliability curve*. We are especially interested in the behavior of the relative failure probability curve in the vicinity of the origin $(0, 0)$, since this region corresponds to highly reliable networks, and an accurate estimation of failure probabilities in this region is especially challenging.

In this paper, we compare the reliability properties of $\otimes(n^2, k+2)$ and $\boxtimes(n, k)$ for two values of n , namely, $n = 5$ and $n = 8$. For each n , we consider several values of k , i.e. several different numbers of random shortcuts per node. The number of samples used in each Subset Simulation run is $N = 2000$ per level, and the conditional failure probability is $p_0 = 0.1$. Figures 2 and 3 show the resulting relative reliability curves for $n = 5$ and $n = 8$, respectively.

For $n = 5$, $k = 0, 1, 2$, and 3 are studied. For both small-world ring and small-world torus models, $M = 200$ network realizations $\{\hat{\otimes}_i\}$ and $\{\hat{\boxtimes}_i\}$ and

source-sink pairs $\{(a_i, b_i)\}$ are generated to estimate the expected failure probabilities in (21) and (22). A fine grid of μ^* -values between 0.1 and 3 is used to obtain the relative reliability curve in Figure 2. For $n = 8$, $k = 0, 1, 2$, and 4 are considered. In this case, $M = 100$ network realizations and source-sink pairs are simulated to estimate $\bar{p}_{\mathcal{F}, \otimes}(\mu^*)$ and $\bar{p}_{\mathcal{F}, \boxtimes}(\mu^*)$ in (21) and (22), respectively. A fine grid of μ^* -values between 0.1 and 6 is used to obtain the relative reliability curve in Figure 3.

For both $n = 5$ and $n = 8$, we observe the following results:

1. The relative reliability curve lies below the equal reliability line, i.e. below the diagonal that connects the origin $(0,0)$ with $(1,1)$. This indicates that, on average, the small-world torus model produces a more reliable (in the max-flow sense) network than the small-world ring model.
2. When k increases, the relative reliability curve shifts towards the equal reliability line. This result is expected because as k increases, both network models become topologically closer to each other (both converge to a random graph), and, therefore, their reliability properties become similar.
3. The relative reliability curve is approximately linear in the log-scale, i.e.

$$\bar{p}_{\mathcal{F}, \boxtimes} \approx (\bar{p}_{\mathcal{F}, \otimes})^\alpha, \quad \alpha < 1 \quad (23)$$

This suggests that when both models produce highly reliable networks, i.e. when the critical threshold μ^* is very small, the small-world torus model produces a substantially more reliable network than the small-world ring model does.

Thus, our simulation results show that the small-world torus model is more efficient in producing reliable networks than the small-world ring model, where reliability is understood as the maximum-flow reliability. It is important to highlight that if direct Monte Carlo simulation were used instead of Subset Simulation for estimating failure probabilities $p_{\mathcal{F}}(\hat{\otimes}_i; (a_i, b_i); \mu^*)$ and $p_{\mathcal{F}}(\hat{\boxtimes}_i; (a_i, b_i); \mu^*)$ in (21) and (22), respectively, then it would take impractically long time to obtain these results.

5 CONCLUSIONS

In this paper, we propose a framework for quantitative assessment of network reliability, formulate a general network reliability problem within this framework, and propose an efficient Markov chain Monte Carlo method for solving this problem. The proposed method is based on Subset Simulation, a very efficient stochastic simulation method originally developed for estimation of small failure probabilities of complex

civil engineering structures such as tall buildings and bridges at risk from earthquakes. The efficiency of the method is demonstrated with an illustrative example where two small-world network models, namely, the small-world torus model and the small-world ring model, are compared in terms of reliability of networks they produce, where reliability is understood as the maximum-flow reliability. Simulation results demonstrate that the small-world torus model is more efficient in producing reliable networks.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under award number EAR-0941374 to the California Institute of Technology. This support is gratefully acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect those of the National Science Foundation.

REFERENCES

- Ahuja, R., T. Magnati, & J. Orlin (1962). *Network flows: theory, algorithms, and applications*. Upper Saddle River, NJ: Prentice Hall.
- Amaral, L., A. Scala, M. Barthelemy, & H. Stanley (2000). Classes of behavior of small-world networks. *Proc. Natl. Acad. Sci.* 97, 11149–11152.
- Au, S. & J. Beck (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Prob Eng Mech* 16, 263–277.
- Ball, M., C. Colbourn, & J. Provan (1995). Network reliability. *Handbooks in OR and MS* 7, 673–762.
- Colbourn, C. (1987). *The combinatorics of network reliability*. New York, USA: Oxford University Press.
- Doob, J. (1953). *Stochastic processes*. New York: Wiley.
- Ford, L. & D. Fulkerson (1962). *Flows in networks*. Princeton, NJ: Princeton University Press.
- Hastings, W. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika* 57, 97–109.
- Jonczy, J. & R. Haenni (2007). A new approach to network reliability. *5rd Int. Conference on Mathematical Methods In Reliability No. 196*.
- Milgram, S. (1967). The small world problem. *Psychology Today* 2, 60–67.
- Robert, C. & G. Casella (2004). *Monte Carlo Statistical Methods*. Springer Texts in Statistics.
- Rosenthal, A. (1977). Computing the reliability of complex networks. *SIAM Journal on Applied Mathematics* 32, 384–393.
- Sen, P., S. Dasgupta, A. Chatterjee, P. Sreeram, G. Mukherjee, & S. Manna (2003). Small-world properties of the indian railway network. *Phys Rev E* 67, 036106.
- Watts, D. & S. Strogatz (1998). Collective dynamics of “small-world” networks. *Nature* 393, 440–442.
- Zio, E. (2007). Reliability analysis of complex network systems: research and practice in need. *IEEE Reliability Society 2007 Annual Technology Report.*
- Zuev, K., J. Beck, S. Au, & L. Katafygiotis (2012). Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. *Computers and Structures* 92–93, 283–296.