Probabilistic Engineering Mechanics 40 (2015) 25–35

Contents lists available at ScienceDirect

Probabilistic Engineering Mechanics

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

General network reliability problem and its efficient solution by Subset Simulation

Konstantin M. Zuev [a,*], Stephen Wu [b], James L. Beck [b]

[a] University of Liverpool, United Kingdom
[b] California Institute of Technology, United States

ARTICLE INFO

Article history:
Received 18 July 2014
Received in revised form
14 January 2015
Accepted 2 February 2015
Available online 4 February 2015

Keywords:
Network reliability
Technological networks
Markov chain Monte Carlo
Subset Simulation
Small-world network models

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 in delivering their service. As a consequence, there is a pressing need for efficient computational methods that can quantitatively assess the reliability of technological networks to enhance their design and operation in the presence of uncertainty in their future demand, supply and capacity. In this paper, we propose a stochastic framework for quantitative assessment of the reliability of network service, formulate a general network reliability problem within this framework, and then show how to calculate the service reliability using Subset Simulation, an efficient Markov chain Monte Carlo method that was originally developed for estimating small failure probabilities of complex dynamic systems. The efficiency of the method is demonstrated with an illustrative example where two small-world network generation models are compared in terms of the maximum-flow reliability of the networks that 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 rail lines, or networks of airline routes), communication networks (e.g. telephone networks or the Internet), and utility networks (e.g. networks for delivery of electricity, gas or water).

Technological networks are so deeply integrated into the infrastructure of megacities that their failures, although rare, often have serious consequences on the wellbeing of society. Societal dependence on technological systems and networks is constantly growing, giving an ever increasing vulnerability to their failure. As a result, there is an increasing demand for modern technological networks to be highly reliable in their operations. The degree to which a network is able to provide the required service needs to be quantitatively assessed during its design and operation, taking into account uncertainty in the future demand, supply and network operational capacity.

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 methods such as minpath or mincut enumeration, e.g. [1]. *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, e.g. [13]. 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, e.g. [21]. A detailed review of traditional methods for reliability analysis of small scale networks is provided in [15]. 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. Fig. 1 shows a visual representation of a small portion

[*] Corresponding author.
E-mail addresses: zuev@liverpool.ac.uk (K.M. Zuev), stewu@caltech.edu (S. Wu), jimbeck@caltech.edu (J.L. Beck).

http://dx.doi.org/10.1016/j.probengmech.2015.02.002
0266-8920/© 2015 Elsevier Ltd. All rights reserved.

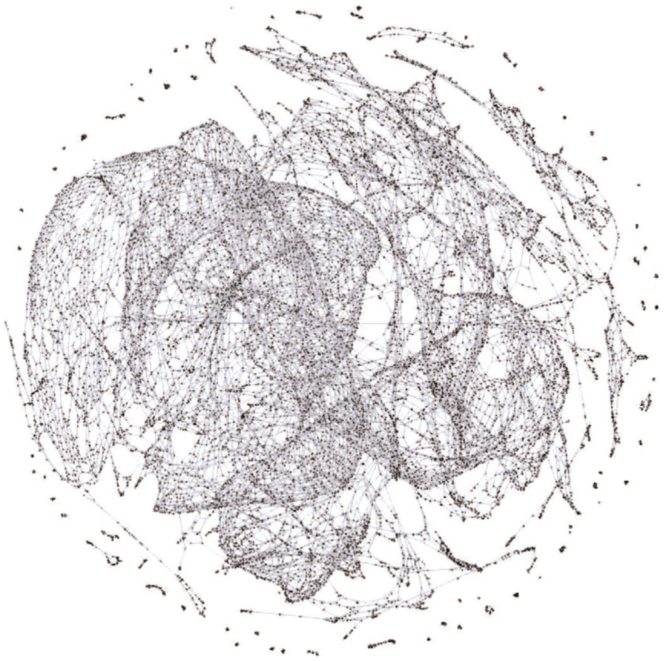


Fig. 1. Man-made "galaxy": a visual representation of a small portion (~1%) of a California road network. Intersections and road endpoints are represented by vertices and the roads connecting these intersections or endpoints are represented by undirected edges. The network data is available for free at http://snap.stanford.edu/data/roadNet-CA.html. Visualization was done using the Network Workbench Tool (available for free at http://nwb.slis.indiana.edu).

(approximately 1%) of a California road network. In this network, intersections and road endpoints are represented by vertices and the roads connecting these intersections or endpoints are represented by undirected edges.

This dramatic change of scale induces 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 [27] 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 [32]. 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 stochastic framework for quantitative assessment of network reliability in the presence of uncertainty, formulate a general network reliability problem within this framework, and show how to solve this problem using Subset Simulation [5], an efficient Markov chain Monte Carlo method that was originally developed for estimating small failure probabilities of complex dynamic systems, such as civil engineering structures at risk from earthquakes. The new theory was first presented in the conference paper [36] but here we give a fuller explanation and extended results. We remark that Subset Simulation has also been used previously for evaluating origin–destination connectivity reliability of lifeline networks [9].

We proceed as follows. In the next section, we highlight the similarity between reliability problems for complex systems and complex networks, and 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. From complex systems to complex networks

Complex networks are often viewed as the structural skeletons of complex dynamic systems. While networks are a relatively new object of study in reliability engineering, the reliability of dynamic systems is a well-established and deeply researched problem. The engineering research community has developed several very efficient methods for estimation of reliability of complex dynamic systems such as tall buildings, bridges, and aircraft [4,5,18,7,8,16,34,35]. Moreover, it can be shown (see, e.g. [33]) that the system reliability problem is mathematically equivalent to two other extensively researched problems: finding the free energy of a physical system (statistical mechanics), and finding the marginal likelihood of a Bayesian statistical model (Bayesian statistics). All three problems can be considered as the problem of estimating the ratio of normalizing constants for a pair of probability distributions.

As a first step towards efficient network reliability methods, this paper focuses on the development of a network analog of the system reliability method, Subset Simulation [5]. In Section 2.1, we briefly review the system reliability problem to demonstrate its similarity with the network reliability problem which is discussed in Section 2.2.

2.1. System reliability problem

Calculation of the reliability, or equivalently the probability of failure p_F, of a dynamic system under given excitation conditions is one of the most important and challenging problems in reliability engineering. The uncertainty in the input excitation $x \in \mathbb{R}^m$ is quantified by a joint probability density function (PDF) $\pi(x)$. The performance of the dynamic system under this input is quantified by a *performance function* $\mu: \mathbb{R}^m \to \mathbb{R}$ through a dynamic input–output model of the system. For example, if our system corresponds to a tall building, the input x may represent an uncertain earthquake excitation sampled at discrete times over some interval and the performance $\mu(x)$ may represent the corresponding maximum roof displacement over this duration, or the maximum interstory drift over all stories for the duration, calculated from the dynamic model.

Define the failure domain $F \subset \mathbb{R}^m$ as the set of inputs ("failure points") that lead to the exceedance of some prescribed critical threshold $\mu^* \in \mathbb{R}$:

$$F = \{x \in \mathbb{R}^m | \mu(x) > \mu^*\} \tag{1}$$

In the above example, the critical threshold μ^* represents the maximum permissible roof displacement or maximum permissible interstory drift and so the failure domain F represents the set of all earthquake excitations that lead to unacceptable deformation of the tall building.

The *system reliability problem* is then to compute the probability of failure that is given by the following integral:

$$p_F = \mathbb{P}(x \in F) = \int_F \pi(x)\,dx = \int_{\mathbb{R}^m} \pi(x) I_F(x)\,dx = \mathbb{E}_\pi[I_F], \tag{2}$$

where $\mathbb{E}_\pi$ denotes expectation with respect to the distribution $\pi(x)$ and I_F is the indicator function of the failure domain F: $I_F(x) = 1$ if the system subject to excitation x fails (i.e. the output $\mu(x)$ is not acceptable according to the performance criterion, $\mu(x) > \mu^*$) and

$I_F(x) = 0$ otherwise.

The following set of realistic assumptions makes this problem computationally very challenging:

S1. *The relationship between x and $I_F(x)$ is not explicitly known*. Although for any x we can check whether it is a failure point or not by numerical analysis using a model of the dynamic system, i.e. calculate the value $I_F(x)$ for a given x, we cannot usually obtain an explicit analytical formula for $\mu(x)$ or its gradient with respect to x.

S2. *The computational effort for the dynamic system analysis needed to evaluate $\mu(x)$ for each value of x is significant so that it is essential to minimize the number of such function evaluations*. Complex dynamic systems (e.g. bridges or aircraft) are represented by complex models. In this context, complexity means, in particular, that the evaluation of $\mu(x)$ for any x is very time-consuming. Thus, it is important to reduce the number of such function evaluations.

S3. *The dimension m is large*, *i.e.* $m \gg 1$, because the stochastic input time history is discretized in time. For example, $m \sim 10^3$ is not unusual in the reliability literature.

S4. *The probability of failure p_F is very small*, *i.e.* $p_F \ll 1$. In other words, the system is assumed to be designed properly, so that failure is a rare event. In the reliability engineering literature, $p_F \sim 10^{-9}$–10^{-2} have been considered.

Due to these conditions, both numerical integration and standard Monte Carlo are computationally infeasible for estimating the high-dimensional integral in (2).

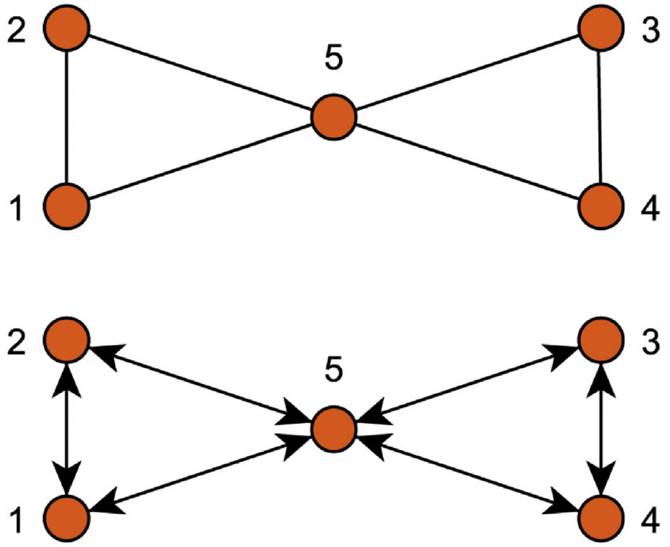
Over the past decade, the engineering research community has realized the importance of advanced stochastic simulation methods for reliability analysis. As a result, many efficient algorithms have been recently developed, e.g. Subset Simulation by Au and Beck [5], Line Sampling by Koutsourelakis et al. [18], Auxiliary Domain method by Katafygiotis et al. [16], Horseracing Simulation by Zuev and Katafygiotis [34], Bayesian Subset Simulation by Zuev et al. [35], to name but a few.

Since complex networks can be viewed as the structural skeletons of complex systems, it is natural to expect that the above techniques can be adapted for efficient estimation of network reliability.

2.2. Network reliability problem

A network topology is represented as a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ 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 directly 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 ith 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. The concept of adjacency matrix and the equivalence of undirected graphs to directed ones are demonstrated in Fig. 2.

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 [10]. Thus, this assumption does not, in fact, limit the generality, and networks with link failures only are sufficiently general.



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Fig. 2. The undirected graph on the top is equivalent to the directed graph at the bottom. Both graphs are described by the adjacency matrix A.

A *network state* is defined as an m-tuple $s = (s_1, \ldots, s_m)$ 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 *network state space* $\mathcal{S}$ is then an m-dimensional hypercube:

$$\mathcal{S} = \left\{(s_1, \ldots, s_m) | s_i \in [0, 1]\right\} = [0, 1]^m \tag{3}$$

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. We write $s \sim \pi(s)$. 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 $\pi(s)$ on $\mathcal{S}$: each state $s = (s_1, \ldots, s_m)$ with $s_i = 0, 1$ has occurrence probability

$$\pi(s) = \prod_{i=1}^{m} q_i^{s_i} p_i^{1 - s_i}, \tag{4}$$

where $q_i = 1 - p_i$ is the link reliability; partially operational links do not occur in this case.

In this work, we focus on the more general case of continuous state spaces where $\pi(s)$ denotes a PDF on $\mathcal{S}$. Furthermore, we define a *performance function* $\mu: \mathcal{S} \to \mathbb{R}$ 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. Similar to the system case (1), the *failure domain* $\mathcal{F} \subset \mathcal{S}$ is defined as follows:

Table 1
Notation.

$G=(V,E)$	Graph representing a network
$V=\{v_1,\ldots,v_n\}$	Set of nodes (or vertices)
$E=\{e_1,\ldots,e_m\}$	Set of links (or edges)
$A(G)$	Adjacency matrix of G
$d(v)$, $d_{in}(v)$, $d_{out}(v)$	Degree, in-degree, and out-degree of node v
$s=(s_1,\ldots,s_m)$	A network state, where $s_i\in[0,1]$, $i=1,\ldots,m$
$\mathcal{S}=[0,1]^m$	Network state space (set of all network states)
$\pi(s)$	Probability distribution on $\mathcal{S}$
$\mu:\mathcal{S}\to\mathbb{R}$	Network performance function
$\mathcal{F}\subset\mathcal{S}$	Failure domain

$$\mathcal{F}=\{s\in\mathcal{S}|\mu(s)<\mu^*\}, \tag{5}$$

where μ^* is the critical threshold. The notation is summarized in Table 1.

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

$$p_{\mathcal{F}}=\mathbb{P}(s\in\mathcal{F})=\int_{\mathcal{S}}\pi(s)I_{\mathcal{F}}(s)\,ds=\mathbb{E}_\pi[I_{\mathcal{F}}]. \tag{6}$$

Several classical reliability problems [6,27,10] 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.

To mirror the systems framework S1–S4, we make the following real-life assumptions:

N1. *The relationship between $s\in\mathcal{S}$ and $I_{\mathcal{F}}(s)$ is not explicitly known.*
N2. *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.
N3. *The number of links m is large, i.e. $m\gg1$.* Many actual networks have millions (e.g. road networks), or even billions, of links (e.g. the Internet).
N4. *The probability of failure $p_{\mathcal{F}}$ is very small, i.e. $p_{\mathcal{F}}\ll1$.* 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 N3, the integral in (6) is taken over a high-dimensional hypercube and, due to N2, 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 (6) renders standard Monte Carlo method theoretically 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 $\pi(s)$:

$$\hat{p}_{\mathcal{F}}^{(MC)}=\frac{1}{N}\sum_{i=1}^{N}I_{\mathcal{F}}(s^{(i)}),\quad s^{(i)}\sim\pi(s) \tag{7}$$

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

$$\delta_{MC}=\sqrt{\frac{1-p_{\mathcal{F}}}{Np_{\mathcal{F}}}} \tag{8}$$

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 ($\sim1/p_{\mathcal{F}}$) 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 N2 and N4, standard Monte Carlo becomes computationally prohibitive for our problems of interest involving small failure probabilities.

The basic strategy that is often employed in this situation is to generate more samples in the "important region" of the failure domain, i.e. in the region $\mathcal{F}'\subseteq\mathcal{F}$ that contains most of the probability mass and, therefore, contributes mostly to the integral (6). More formally, we want to sample from distribution $\psi(s)$ on $\mathcal{F}'$, where $\psi(s)$ is the conditional distribution $\psi(s)=\pi(s|\mathcal{F}')$ or, possibly, some other importance sampling distribution with support $\operatorname{supp}\psi(s)=\mathcal{F}'$. To achieve this goal, we can use *Markov chain Monte Carlo* (MCMC) methods, a class of algorithms for sampling from complex probability distributions. In MCMC, samples from a given (up to a normalizing constant) distribution $\psi(s)$ are generated by simulating a Markov chain whose state distribution converges to the target distribution $\psi(s)$ as its stationary distribution. Markov chain Monte Carlo sampling originated in statistical physics, and now is widely used in solving statistical problems [20,26]. In the next section we show how MCMC can be employed for solving the network reliability problem.

3. Subset Simulation

The main idea of Subset Simulation [5] 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:

$$\mathcal{S}=\mathcal{F}_0\supset\mathcal{F}_1\supset\cdots\supset\mathcal{F}_L=\mathcal{F} \tag{9}$$

Subsets $\mathcal{F}_0,\ldots,\mathcal{F}_{L-1}$ are called *intermediate failure domains*. They are schematically shown in Fig. 3. The failure probability can be written then as a product of conditional probabilities:

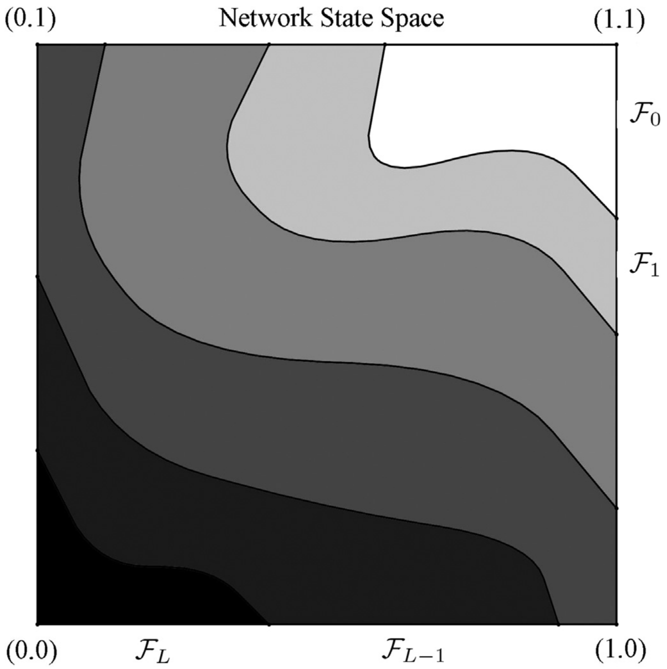


Fig. 3. A schematic representation of the intermediate failure domains $\mathcal{S}=\mathcal{F}_0\supset\mathcal{F}_1\supset\cdots\supset\mathcal{F}_L=\mathcal{F}$ in the network state space $\mathcal{S}=[0,1]^m$, where $m=2$.

$$p_{\mathcal{F}} = \prod_{j=1}^{L} \mathbb{P}(\mathcal{F}_j|\mathcal{F}_{j-1}) = \prod_{j=1}^{L} p_j, \tag{10}$$

where $p_j = \mathbb{P}(\mathcal{F}_j|\mathcal{F}_{j-1})$ is the conditional probability at the $(j-1)$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):

$$p_1 \approx \hat{p}_1 = \frac{1}{N}\sum_{i=1}^{N} I_{\mathcal{F}_1}(s_0^{(i)}), \quad s_0^{(i)} \overset{i.i.d.}{\sim} \pi(s|\mathcal{F}_0) \equiv \pi(s) \tag{11}$$

We assume here that $\mathcal{F}_1$ is chosen in such a way that p_1 is relatively large, so that the MCS estimate (11) is accurate for a moderate sample size N. Later in this section, we will discuss how to chose intermediate failure domains $\mathcal{F}_j$ adaptively.

For $j \ge 2$, to estimate p_j using MCS one needs to simulate i.i.d. samples from the 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 $j \ge 2$ can be done by a specifically tailored MCMC technique at the expense of generating dependent samples.

The Metropolis–Hastings (MH) algorithm [22,14] 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 at level (j−1) and $q(s|s_{j-1}^{(i)})$, called the proposal PDF, be an m-dimensional PDF on the network state space $\mathcal{S}$ that may depend on $s_{j-1}^{(i)}$ and can be readily sampled. Then the MH update $s_{j-1}^{(i)} \to s_{j-1}^{(i+1)}$ 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 $\alpha = I_{\mathcal{F}_{j-1}}(\xi)\min\left\{1, \pi(\xi)q(s_{j-1}^{(i)}|\xi)/\pi(s_{j-1}^{(i)})q(\xi|s_{j-1}^{(i)})\right\}$.
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* [26] and Subset Simulation has this property because of the way the seeds are selected.

It was observed in [5] 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) [5], a component-wise 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 component ξ_k of the candidate state vector ξ is generated separately using a univariate proposal distribution $q_k(s_k|s_{j-1,k}^{(i)})$ dependent on the kth component $s_{j-1,k}^{(i)}$ of the current state sample $s_{j-1}^{(i)}$. 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 [5] and to [35] 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 (11), where MCS samples are replaced by MCMC samples:

$$p_j \approx \hat{p}_j = \frac{1}{N}\sum_{i=1}^{N} I_{\mathcal{F}_j}(s_{j-1}^{(i)}), \quad s_{j-1}^{(i)} \overset{MMA}{\sim} \pi(s|\mathcal{F}_{j-1}) \tag{12}$$

Note that all samples $s_{j-1}^{(1)}, \dots, s_{j-1}^{(N)}$ in (12) 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 [11].

Subset Simulation uses the estimates (11) for p_1 and (12) for p_j, $j \ge 2$, to obtain the estimate for the failure probability:

$$p_{\mathcal{F}} \approx \hat{p}_{\mathcal{F}} = \prod_{j=1}^{L} \hat{p}_j \tag{13}$$

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 $\mathcal{F} = \{s \in \mathcal{S}|\mu(s) < \mu^*\}$. The sequence of intermediate failure domains can then be defined in a similar way:

$$\mathcal{F}_j = \{s \in \mathcal{S}|\mu(s) < \mu_j^*\}, \tag{14}$$

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 (15) below) so that the estimated conditional probabilities are equal to a fixed value $p_0 \in (0,1)$. In [35], p_0 is called the conditional failure probability.

Subset Simulation algorithm 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 $s_0^{(1)}, \dots, s_0^{(N)} \overset{i.i.d.}{\sim} \pi(s)$
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 $N_{\mathcal{F}}(j) < Np_0$ do
    $j \leftarrow j + 1$
    Sort $\{\mu^{(i)}\}$: $\mu^{(i_1)} \le \mu^{(i_2)} \le \dots \le \mu^{(i_N)}$
    Define the $j$th intermediate critical threshold:
    $\mu_j^* = \frac{\mu^{(i_{Np_0})} + \mu^{(i_{Np_0+1})}}{2}$        (15)
    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 $p_{\mathcal{F}}$:
$\hat{p}_{\mathcal{F}} = p_0^j \frac{N_{\mathcal{F}}(j)}{N}$        (16)
```

The adaptive choice of μ_j^* – values in (15) 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 [35] 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. Illustrative example: maximum flows in small-world networks

In this section, we demonstrate how Subset Simulation can be used for efficient solution of the *maximum-flow reliability problem* for networks generated from *small-world models*.

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 [12].

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 flow *capacity* $s(v, u) \ge 0$. A quadruple $(G, a, b, \{s\})$ is often referred to as a *flow network*. A *flow* on G is non-negative function $f: E \to \mathbb{R}^+$ that satisfies the following properties:

- *Capacity constraint*: the flow along any link cannot exceed the capacity of that link:
$$f(v, u) \le s(v, u) \quad \text{for all } (v, u) \in E \tag{17}$$
- *Flow conservation*: the total flow entering node v must equal the total flow leaving v for all nodes except a and b:
$$\sum_{u \in V} f(u, v) = \sum_{u \in V} f(v, u) \quad \text{for all } v \in V \backslash \{a, b\} \tag{18}$$
For convenience, it is assumed in (18) that $f(v, u) = 0$ if $(v, u) \notin E$ (no link from node v to node u).

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

$$|f| = \sum_{v \in V} f(a, v) - \sum_{v \in V} f(v, a) \tag{19}$$

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

$$|f| = \sum_{v \in V} f(v, b) - \sum_{v \in V} f(b, v) \tag{20}$$

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 $f^\star = \arg\max |f|$ over all possible flows f in a given flow network $(G, a, b, \{s\})$. 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 [2].

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, $0 \le s_i \le 1$, and suppose that, instead of being prescribed, they are uncertain with their uncertainty quantified by a probability model π for all link capacities, i.e. a probability distribution on the network state space $\mathcal{S} = \{s = (s_1, \dots, s_m) | 0 \le s_i \le 1\}$. 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:

$$\mu_{MF}(s) = |f^\star(s)| \tag{21}$$

The failure domain $\mathcal{F} \subset \mathcal{S}$ can now be defined as before, $\mathcal{F} = \{s \in \mathcal{S} | \mu_{MF}(s) < \mu^*\}$, where μ^* is the critical threshold. Thus, the introduction of uncertain link capacities brings us into the general network reliability framework described in Section 2.2.

One of the fundamental questions in network science and reliability theory is the following: given n nodes and m links, how should they be combined into the most reliable network? This is a computationally challenging optimization problem. In the next section we analyze a simpler but related question: given two network models that generate componentwise equivalent but topologically different networks, how do we find out which network model produces more reliable networks?

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 [23]. 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 [30,24], including technological ones such as power grids [31], airline routes [3], and railways [28].

In their seminal paper [31], Watts and Strogatz 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 in 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. Other small-world network models that have become popular include the Newman–Watts model [25], where, instead of rewiring links, new links connecting randomly chosen pairs of nodes are added, and the Kleinberg model [17], where small-world networks are built on a d-dimensional lattice and the probability that two nodes are connected by a long-range link depends on the distance between them in the lattice. In general, small-world models can be constructed on lattices of any dimension and any topology. For a state-of-the-art review of the subject, we refer the reader to [24].

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 with 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 (recall Fig. 2). Next, for each node v, the model generates k additional directed links connecting that node with k other nodes $v_1, \ldots, v_k$ chosen uniformly at random. Note that for each selected v_i, either link (v, v_i) or (v_i, v) is constructed 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.

Small-world network models are often used to study practical problems in various fields, such as transportation, biology, and social science [19]. For example, [29] reviews recent research that has been done in social science and management using small-world networks. Another important application is the reliability study of transportation networks. The development of a transportation network usually starts with a fixed plan to connect some

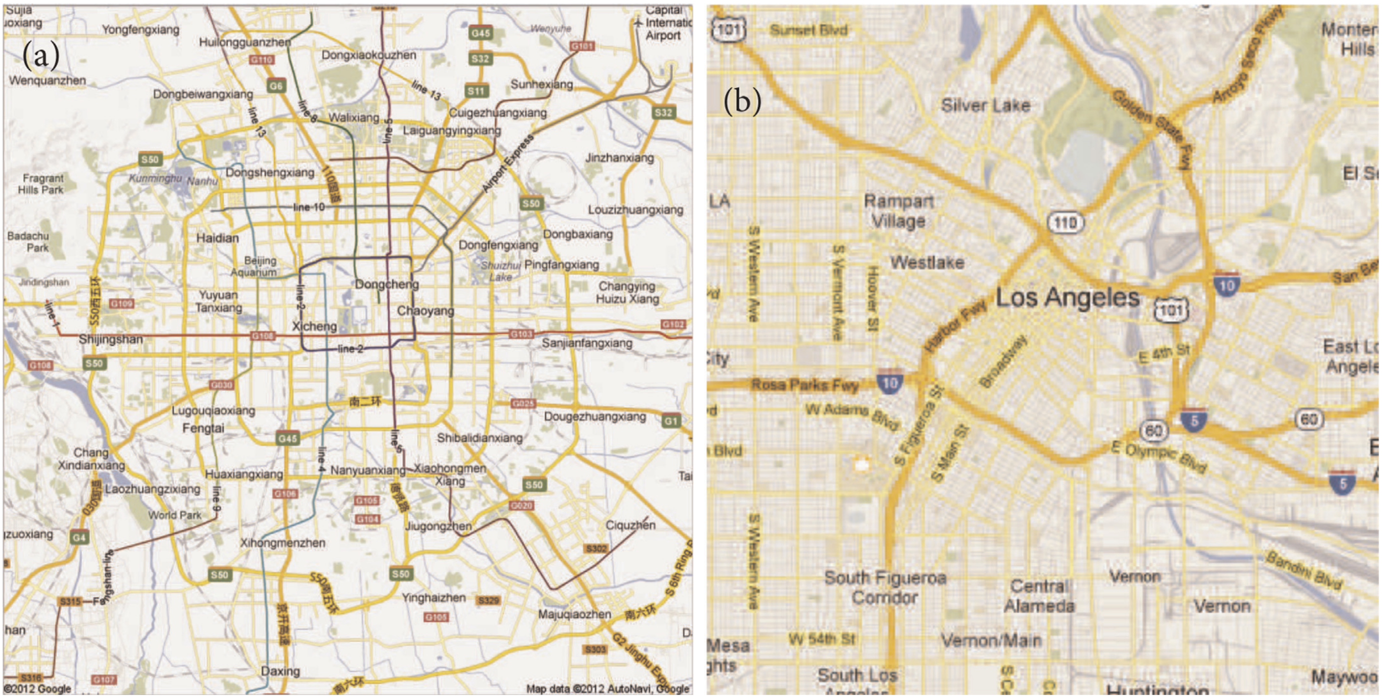


Fig. 4. Road networks in (a) Beijing, China and (b) Los Angeles, USA. Both plots are obtained from Google Maps. The Beijing and LA networks can be modeled by the ring model (like Fig. 5(a)) and the torus model (like Fig. 5(b)), respectively.

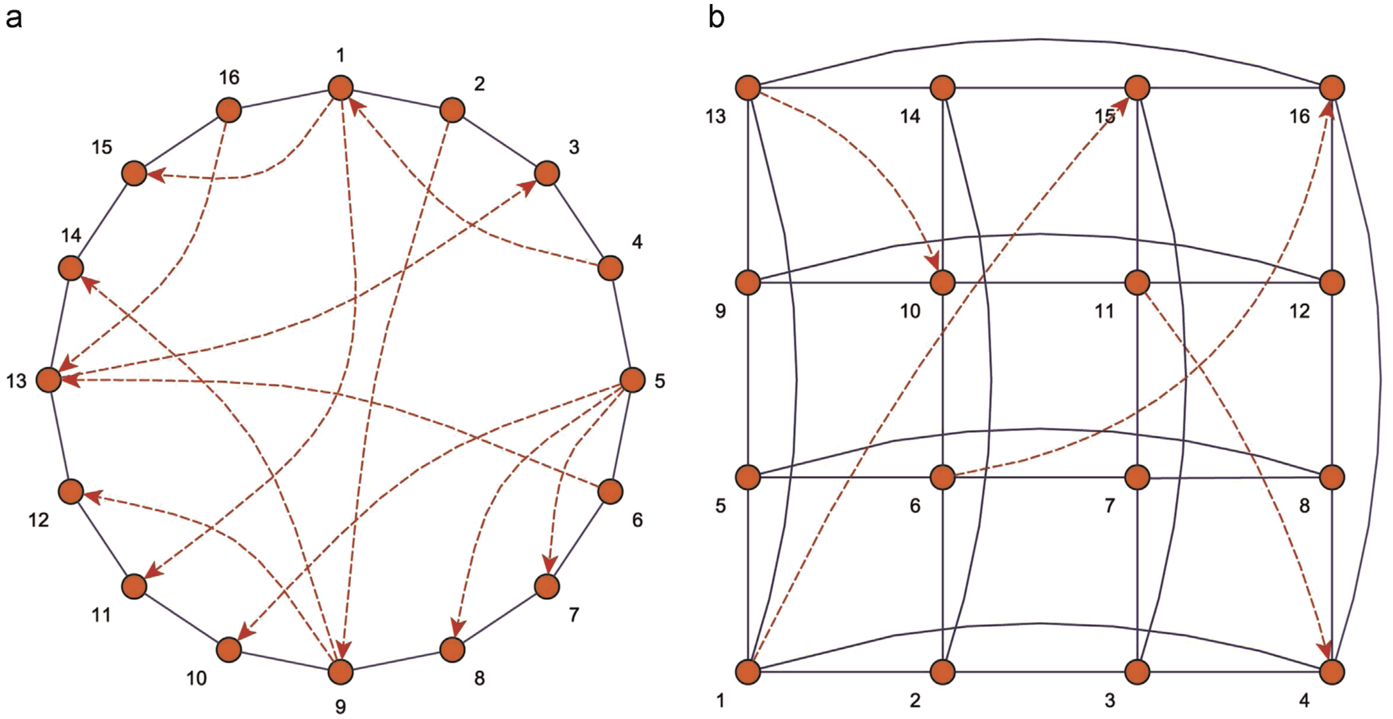


Fig. 5. (a) Realization of the small-world ring model $\otimes(n, k)$, with $n=16$ and $k=3$. Solid and dashed lines represent regular links and random shortcuts, respectively. For visibility, only random shortcuts that correspond to nodes 1, 5, 9, and 13 are shown. (b) Realization of the small-world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

regions in a city, and then more roads are built as the city grows and the demand of roads increases. The small-world network resembles this process very well. The similarity between real transportation networks and synthetic realizations of small-world models can often be visually observed. For example, Fig. 4(a) shows part of the transportation network in Beijing, China, that can be represented by the small-world ring model; Fig. 4(b) shows part of the transportation network in Los Angeles, USA, that can be represented by the small-world torus model.

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 Fig. 5(a) and (b), respectively.

In Table 2, the total number of network components is provided for both models. 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

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

$$\text{Topologically:} \quad \otimes(n^2, k+2) \neq \boxtimes(n, k) \tag{22}$$

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, which represents the tradeoff between local connectivity and global reachability, has the potential to yield significantly different reliability properties for the two network models. Fig. 6(a) and (b) shows an example for both models when n is large. To demonstrate the effectiveness of Subset Simulation for network reliability estimation, we investigate the following question: Which model, $\otimes(n^2, k+2)$ or $\boxtimes(n, k)$, produces more reliable networks where reliability is understood as the maximum-flow reliability?

Table 2
Componentwise comparison of two small-world models, $\otimes(n, k)$ and $\boxtimes(n, k)$.

Model	# of nodes	# of regular links	# of random shortcuts	Total # of links
$\otimes(n, k)$	n	$2n$	kn	$n(k+2)$
$\boxtimes(n, k)$	n^2	$4n^2$	kn^2	$n^2(k+4)$

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 (21), 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:

$$\bar{p}_{\mathcal{F},\otimes}(\mu^*) = \mathbb{E}_{\otimes,(a,b)}[p_{\mathcal{F}}(\hat{\otimes}; (a, b); \mu^*)] \approx \frac{1}{M}\sum_{i=1}^{M} p_{\mathcal{F}}(\hat{\otimes}_i; (a_i, b_i); \mu^*) \tag{23}$$

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

$$\bar{p}_{\mathcal{F},\boxtimes}(\mu^*) = \mathbb{E}_{\boxtimes,(a,b)}[p_{\mathcal{F}}(\hat{\boxtimes}; (a, b); \mu^*)] \approx \frac{1}{M}\sum_{i=1}^{M} p_{\mathcal{F}}(\hat{\boxtimes}_i; (a_i, b_i); \mu^*), \tag{24}$$

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

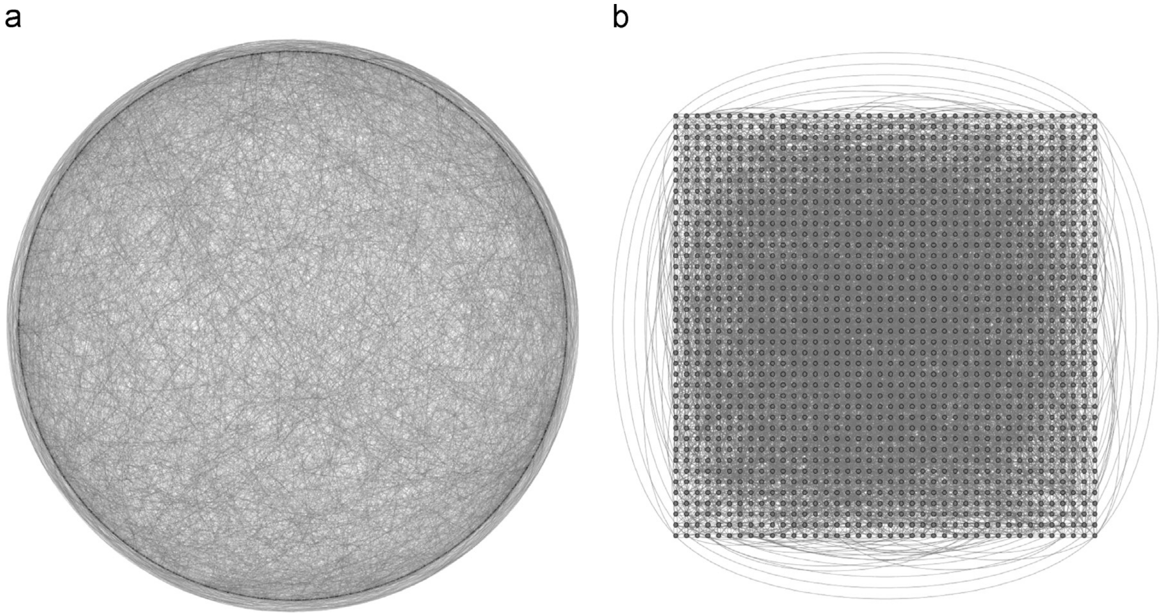


Fig. 6. (a) Realization of the small-world ring model $\otimes(n^2, k+2)$, with n=40 and k=2. (b) Realization of the small-world torus model $\boxtimes(n, k)$, with n=40 and k=2.

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 $\mu^* \to -\infty$; and it ends at (1, 1), since both probabilities converge to 1, as $\mu^* \to +\infty$. 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. Figs. 7 and 8 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 (23) and (24). A fine grid of μ^* – values between 0.1 and 3 is used to obtain the relative reliability curve in Fig. 7. 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 (23) and (24), respectively. A fine grid of μ^* – values between 0.1 and 6 is used to obtain the relative reliability curve in Fig. 8.

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)

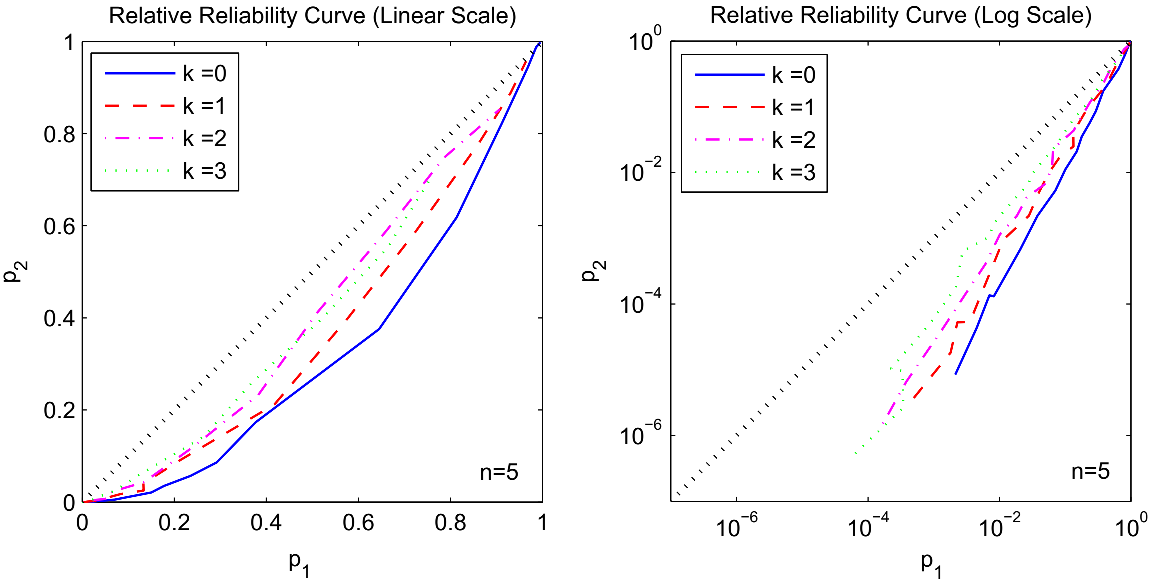


Fig. 7. The relative reliability curve in the linear scale (left panel) and log scale (right panel), for n=5. Here, $p_1 = \bar{p}_{\mathcal{F},\otimes}$ and $p_2 = \bar{p}_{\mathcal{F},\boxtimes}$ as in (23) and (24) for $\mu^* \in (0.1, 3)$.

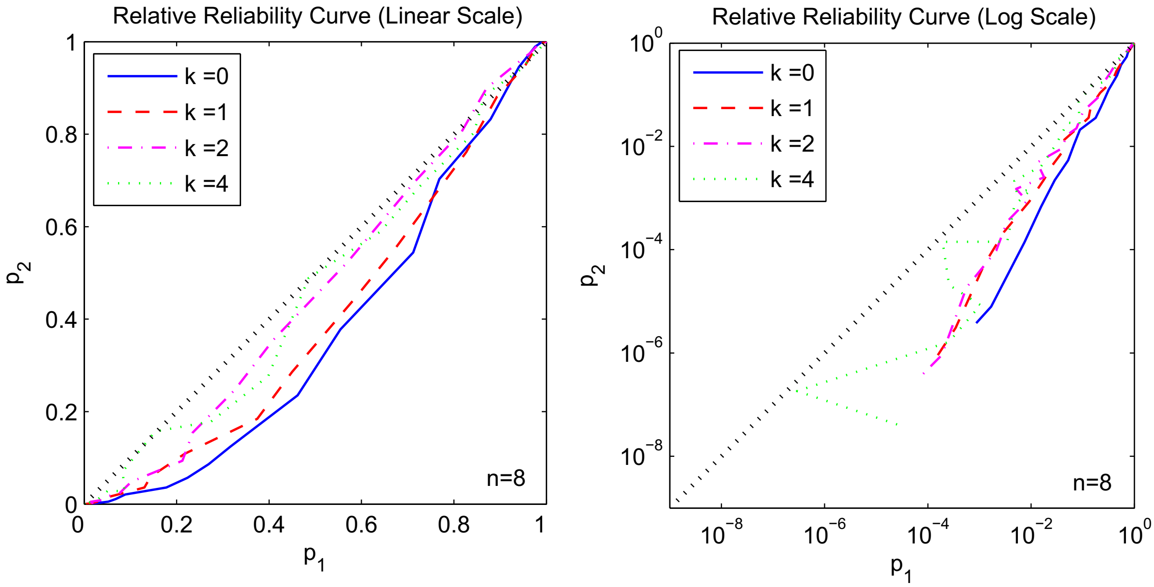


Fig. 8. The relative reliability curve in the linear scale (left panel) and log scale (right panel), for n=8. Here, $p_1 = \bar{p}_{\mathcal{F},\otimes}$ and $p_2 = \bar{p}_{\mathcal{F},\boxtimes}$ as in (23) and (24) for $\mu^* \in (0.1, 6)$.

with (1,1). This indicates that, on average, the small-world torus model produces more reliable (in the max-flow sense) networks 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.

$$\bar{p}_{\mathcal{F},\boxtimes} \approx (\bar{p}_{\mathcal{F},\otimes})^{\alpha}, \quad \alpha > 1 \tag{25}$$

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 substantially more reliable networks 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 the efficiency of a model in generating reliable networks depends on the definition of reliability and, therefore, on the underlying network performance function. As shown above, for the max-flow network performance function μ_{MF} in (21), the torus model is more efficient. However, for some other performance functions, the ring model may turn out to be more efficient. Consider, for example, the average distance between distinct nodes as a measure of network performance:

$$L = \frac{2}{n(n-1)} \sum_{i \neq j} \text{dist}(v_i, v_j), \tag{26}$$

where $\text{dist}(v_i, v_j)$ is the length of the shortest path between v_i and v_j, and n is the number of nodes in the network. It is well known that in small-world networks, L scales logarithmically with the network size [30], i.e. $L \propto \log n$. For the small-world ring model

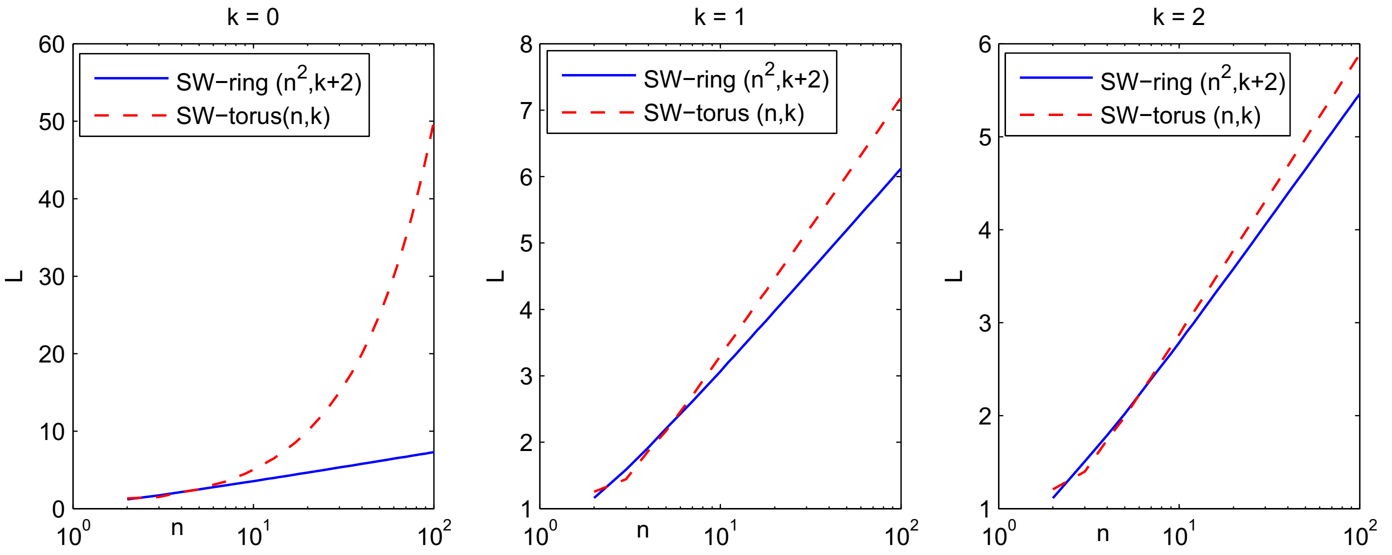


Fig. 9. The expected average distance between distinct nodes L for two network models as a function of network size n for different numbers of random shortcuts k = 0, 1, and 3.

$\otimes(n^2, k+2)$ and small-world torus model $\boxtimes(n, k)$, the estimated values of L are shown in Fig. 9 for different values of n and k. It follows from this plot that, if $n > 10$, then, on average, the ring model produces networks with smaller average distance between nodes, and, thus, it is more efficient in this sense than the torus model.

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 show how to solve this problem using Subset Simulation, an efficient Markov chain Monte Carlo method for rare event simulation that was originally developed for estimating small failure probabilities of complex dynamic systems. The efficiency of the method is demonstrated with an illustrative example where 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 suggest that the small-world torus model is more efficient in producing reliable networks.

Acknowledgments

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

[1] J.A. Abraham, An improved algorithm for network reliability, IEEE Trans. Reliab. 28 (1) (1979) 58–61.
[2] R.K. Ahuja, T.L. Magnati, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Upper Saddle River, NJ, 1993.
[3] L.A.N. Amaral, A. Scala, M. Barthélémy, H.E. Stanley, Classes of behavior of small-world networks, Proc. Natl. Acad. Sci. 97 (2000) 11149–11152.
[4] S.K. Au, J.L. Beck, First-excursion probabilities for linear systems by very efficient importance sampling, Probab. Eng. Mech. 16 (3) (2001) 193–207.
[5] S.K. Au, J.L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probab. Eng. Mech. 16 (4) (2001) 263–277.
[6] M.O. Ball, C.J. Colbourn, J.S. Provan, Network reliability, in: Handbook of Operations Research: Network Models, vol. 7, Elsevier, North-Holland, Amsterdam, 1995, pp. 673–762.
[7] J. Ching, S.K. Au, J.L. Beck, Reliability estimation of dynamical systems subject to stochastic excitation using subset simulation with splitting, Comput. Methods Appl. Mech. Eng. 194 (12–16) (2005) 1557–1579.
[8] J. Ching, J.L. Beck, S.K. Au, Hybrid subset simulation for reliability estimation of dynamic systems subject to stochastic excitation, Probab. Eng. Mech. 20 (2005) 199–214.
[9] J. Ching, W.-C. Hsu, An efficient method for evaluating origin–destination connectivity reliability of real-world lifeline networks, Comput.-Aided Civ. Infrastruct. Eng. 22 (2007) 584–596.
[10] C.J. Colbourn, The Combinatorics of Network Reliability, Oxford University Press, New York, USA, 1987.
[11] J.L. Doob, Stochastic Processes, Wiley, New York, 1953.
[12] L.R. Ford, D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.
[13] N.K. Goyal, Network reliability evaluation: a new modeling approach, in: International Conference on Reliability and Safety Engineering, INCRESE2005, 2005, pp. 473–488.
[14] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97–109.
[15] J. Jonczy, R. Haenni, A new approach to network reliability, in: 5th International Conference on Mathematical Methods in Reliability, MMR'07, No. 196, 2007.
[16] L.S. Katafygiotis, T. Moan, S.H. Cheung, Auxiliary domain method for solving multi-objective dynamic reliability problems for nonlinear structures, Struct. Eng. Mech. 25 (3) (2007) 347–363.
[17] J.M. Kleinberg, Small-world phenomena and the dynamics of information, in: Advances in Neural Information Processing Systems (NIPS), vol. 14, MIT Press, Cambridge, MA, 2001.
[18] P. Koutsourelakis, H.J. Pradlwarter, G.I. Schuëller, Reliability of structures in high dimensions, part I: algorithms and applications, Probab. Eng. Mech. 19 (4) (2004) 409–417.
[19] V. Latora, M. Marchiori, Efficient behavior of small-world networks, Phys. Rev. Lett. 87 (19) (2001) 198701.
[20] J.S. Liu, Monte Carlo Strategies in Scientific Computing, in: Springer Series in Statistics, 2001.
[21] C. Lucet, J. Manouvrier, Exact methods to compute network reliability, in: 1st International Conference on Mathematical Methods in Reliability, MMR'97, 1997, Romania.
[22] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[23] S. Milgram, The small world problem, Psychol. Today 2 (1) (1967) 60–67.
[24] M.E.J. Newman, Networks: An Introduction, Oxford University Press, Oxford, 2010.
[25] M.E.J. Newman, D.J. Watts, Renormalization group analysis of the small-world network model, Phys. Lett. A 263 (1999) 341–346.
[26] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, 2nd ed., in: Springer Texts in Statistics, 2004.
[27] A. Rosenthal, Computing the reliability of complex networks, SIAM J. Appl. Math. 32 (2) (1977) 384–393.
[28] P. Sen, S. Dasgupta, A. Chatterjee, P.A. Sreeram, G. Mukherjee, S.S. Manna, Small-world properties of the Indian railway network, Phys. Rev. E 67 (2003) 036106.
[29] B. Uzzi, L.A.N. Amaral, F. Reed-Tsochas, Small-world networks and management science research: a review, Eur. Manag. Rev. 4 (2007) 77–91.
[30] D.J. Watts, Small Worlds: The Dynamics of Networks between Order and Randomness, Princeton University Press, Princeton, NJ, 1999.
[31] D.J. Watts, S.H. Strogatz, Collective dynamics of small-world networks, Nature 393 (1998) 440–442.
[32] E. Zio, Reliability Analysis of Complex Network Systems: Research and Practice in Need, IEEE Reliability Society Annual Technology Report, 2007.
[33] K.M. Zuev, L.S. Katafygiotis, Estimation of small failure probabilities in high dimensions by adaptive linked importance sampling, in: Computational Methods in Structural Dynamics and Earthquake Engineering, Rethymno, Greece, 2007.
[34] K.M. Zuev, L.S. Katafygiotis, Horseracing simulation algorithm for evaluation of small failure probabilities, Probab. Eng. Mech. 26 (2) (2011) 157–164.
[35] K.M. Zuev, J.L. Beck, S.K. Au, L.S. Katafygiotis, Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions, Comput. Struct. 92–93 (2012) 283–296.
[36] K.M. Zuev, S. Wu, J.L. Beck, Network reliability problem and its efficient solution by subset simulation, in: 11th International Conference on Structural Safety and Reliability, ICOSSAR-2013, 2013.