Technische Universität München
TUM School of Engineering and Design

Adaptive Monte Carlo methods for network reliability assessment

Jianpeng Chan

Vollständiger Abdruck der von der TUM School of Engineering and Design zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Priv.-Doz. Dr.-Ing. Iason Papaioannou

Prüfende der Dissertation:

1. Prof. Dr. Daniel Straub
2. Prof. Jianbing Chen, Ph.D.
3. Prof. Leonardo Duenas-Osori, Ph.D.

Die Dissertation wurde am 18.12.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 08.04.2024 angenommen.

Abstract

Critical infrastructure networks such as water supply systems and power grids are vital for society, and their failure can lead to severe consequences. Due to the uncertainty inherent in both the network performance and external demands and disturbances, the network failure is a random event. Optimal management of the safety of these networks thus benefits from accurately calculating the probability of failure. This is a challenging task. For elementary performance metrics, such as connectivity and maximum flow, the exact calculation of the failure probability is considered to be infeasible within polynomial time constraints, so one must turn to practically efficient algorithms and various approximation techniques, either deterministic or stochastic. For physics-driven metrics, e.g., the blackout size of a power grid, the problem is even more intricate, frequently involving computationally demanding simulation models, while the system performance may not be coherent. Also, the failure probability of critical networks is notably small as they are engineered systems with components designed to prioritize safety. These characteristics pose additional challenges to network reliability assessment, restrict the use of analytical methods, and motivate the use and further development of sampling-based algorithms.

The primary goal of this thesis is to devise efficient sampling-based algorithms for assessing network reliability, with a specific focus on their application to power grids. In this regard, we expand two widely employed variance reduction techniques, namely subset simulation and cross-entropy-based importance sampling, to take into account potential 'jumps' in network performance and to circumvent the overfitting issue when updating discrete parametric models in cross-entropy-based importance sampling (CE-IS). Specifically, we propose a robust adaptation strategy of the intermediate levels within subset simulation and develop efficient Markov Chain Monte Carlo samplers in discrete space. This novel approach is termed adaptive effort subset simulation (aE-SuS). For cross-entropy-based importance sampling, we identify its overfitting issue when updating the categorical distribution and categorical mixture model, and circumvent this issue by incorporating prior information with a Bayesian estimator. This estimator is subsequently integrated into the improved cross entropy (iCE) method for estimating rare events, resulting in the unbiased Bayesian improved cross entropy (BiCE) method. The adaptation of intermediate distributions in BiCE is further substantiated by two theorems for the ideal case, where both the sample size and the capacity of the parametric model are assumed to be infinite. Both aE-SuS and BiCE perform better than state-of-the-art network reliability techniques, e.g., multi-level splitting with the creation process. Critical network components, which contribute significantly to network reliability, can also be identified as a by-product of these algorithms by estimating the component importance measures using failure samples.

Sampling-based methods are applicable to general systems including those utilizing black-box performance functions, yet require many samples to achieve a high accuracy. To compare the efficiency of different sampling-based methods and further explore their potential combinations, we establish unified benchmarks tailored for rare events estimation in power grids and propose a novel relative efficiency metric. The reference failure probabilities can be obtained from principled Monte Carlo methods such as Gamma Bernoulli approximation scheme (GBAS).

Acknowledgements

First and foremost, I want to thank my advisors, Professors Iason Papaioannou and Daniel Straub, for their invaluable support throughout my Ph.D. journey. Their guidance has significantly contributed to my personal growth, both in language proficiency and research skills. Regular discussions with them have deepened my understanding of adaptive Monte Carlo methods and sparked new ideas in this field. Their leadership of the ERA group has created a welcoming academic home for all members, which is truly beneficial.

I would like to extend my appreciation to Professor Leonardo Dueñas-Osorio for hosting me in his group at Rice University. This experience provided me with a deeper understanding of network reliability problems.

I can still clearly remember the day I first arrived in Germany. I was excited, curious, and to be honest, a bit worried about fitting into the new environment. It turns out that my worry is unnecessary, thanks to my colleagues and friends. It is their support that makes most of my time here enjoyable and colourful. Thank you all: Hugo, Mara, Felipe, Eli, Oindrila, Marco, Antonis, Sebastian, Max E, Max T, Daniel K, Daniel H, Amelie, Carmen, Shun, Zeyu, Xiangnan, Bowen, and many others.

Lastly, I want to express my gratitude to my parents, and my girlfriend, Yuan. There is no doubt that their love and patience are what make me fearless and resilient whenever facing challenges throughout this journey. I love you all.

Jianpeng Chan, 19.04.2024

List of publications

Jianpeng Chan is the first author of the following published or submitted journal articles:

- J. Chan, I. Papaioannou, and D. Straub. "An adaptive subset simulation for system reliability analysis with discontinuous limit states". In: *Reliability Engineering & System Safety* 225 (2022), p. 108607.
- J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method for network reliability assessment". In: *Structural Safety* 103 (2023), p. 102344.
- J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method with categorical mixture models".(Under review).
- J. Chan, R. Paredes, I. Papaioannou, L. Duenas-Osorio, and D. Straub. "Adaptive Monte Carlo methods for estimating rare events in power grids". (Under review).

Jianpeng Chan is the second author of the following journal article:

- K. Zwirglmaier, J. Chan, I. Papaioannou, J. Song, and D. Straub. "Hybrid Bayesian networks for reliability assessment of infrastructure systems". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering.* (Accepted).

Jianpeng Chan is the first author of the following published conference articles:

- J. Chan, I. Papaioannou, and D. Straub. "An adaptive subset simulation for system reliability analysis with discontinuous limit states". In: *International Probabilistic Workshop.* Springer. 2021, pp.123-134.
- J. Chan, I. Papaioannou, and D. Straub. "Improved cross-entropy-based importance sampling for network reliability assessment". In: *Proceedings of the 13th International Conference on Structural Safety & Reliability.* ICOSSAR. 2022.
- J. Chan, R. Paredes, I. Papaioannou, L. Duenas-Osorio, and D. Straub. "A comparative study on adaptive Monte Carlo methods for network reliability assessment". In: *Proceedings of the 14th International Conference on Application of Statistics and Probability in Civil Engineering.* ICASP. 2023.

Contents

Part I

Compendium

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

MCS	Monte Carlo simulation
adaptMCS	adaptive Monte Carlo simulation
SuS	Subset simulation
aE-SuS	Adaptive effort subset simulation
CE	Cross entropy
iCE	Improved cross entropy
BiCE	Bayesian improved cross entropy
AMS	Adaptive multi-level splitting
BDD	Binary decision diagram
MMDD	Multi-valued decision diagram
ROBDD	reduced and ordered binary decision diagram
γ-MC	γ-minimal cut
γ-MP	γ-minimal path
SSD	State space decomposition
CNF	Conjunctive normal form
PAC	Probably approximately correct
pMCS	Permutation Monte Carlo simulation
PMF	Probability mass function
PDF	Probability density function
CDF	Cumulative distribution function
FS	Fixed splitting
FE	Fixed effort splitting
IS	Importance sampling
AZIS	Approximate zero-variance importance sampling
KL	Kullback-Leibler
PDEM	Probability density evolution method
MCMC	Markov chain Monte Carlo

Symbols

n	Dimension of the problem, often representing the number of network edges in this thesis
$\boldsymbol{X} = (X_1, \cdots, X_n)$	Input random vector
$\boldsymbol{x} \in \Omega_{\boldsymbol{X}}$	Realization of $\boldsymbol{X}$ that takes value from the set Ω_X
$p_{\boldsymbol{X}}(\boldsymbol{x})$	Input distribution (For discrete input random variables, $p_{\boldsymbol{X}}(\boldsymbol{x})$ is the probability mass function)
$g(\cdot)$	Network performance function
γ	Network performance threshold
F	Failure domain
$p_f \triangleq \Pr(F)$	Failure probability
$\mathcal{G}(\mathcal{V}, \mathcal{E})$	Network with node set $\mathcal{V} \triangleq \{v_1, \cdots, v_m\}$ and edge set $\mathcal{E} \triangleq \{e_1, \cdots, e_n\}$
$\mathcal{K}$	Index set of terminal nodes
K	Number of terminal nodes in the network
$g^{(\text{conn})}$	Network performance function that evaluates the connectivity among terminal nodes
p_d	Failure probability of the d-th network component, $d = 1, \cdots, n$ (The subscript d can be replaced by i, j or k)
$g^{(\text{mf})}$	Network performance function that calculates the maximum flow that can be delivered from the source to the sink
n_d	Number of potential states, such as flow capacities, of the d-th network component
n_{sys}	Number of potential states of network performance
$\boldsymbol{s}_d \triangleq \{s_{d,1}, \cdots, s_{d,n_d}\}$	Potential states, such as flow capacities, of the d-th network component with $0 = s_{d,1} < \cdots < s_{d,n_d}$
$\boldsymbol{p}_d \triangleq \{p_{d,1}, \cdots, p_{d,n_d}\}$	The probability mass function for the states of the d-th network component, $\{s_{d,1}, \cdots, s_{d,n_d}\}$
$g^{(\text{bz})}$	Network performance function that calculates the blackout size of a power grid
$ite(x, G, H)$	If-then-else function that directs to the Boolean function G if the Boolean decision variable x is true and to the Boolean function H otherwise
$\vee$	Logic OR operation
$\wedge$	Logic AND operation
$\neg$	Logic NOT operation
$\Diamond$	Placeholder that can represent either logic OR or logic AND operation
$G_{x=1}$(resp. $G_{x=0}$)	Boolean function G conditional on $x = 1$ (resp. $x = 0$), where 1 denotes true and 0 denotes false
$\boldsymbol{c} = (c_1, \cdots, c_n)$	Maximum lower vector, an extension of the minimal cut-set tailored for coherent multi-state-component systems

e_d	d-th edge of the network
γ-MC	γ-minimal cut, which is referred to as the maximum lower vector in two-terminal maximum flow analysis
N_c	Number of all maximum lower vectors
A_i	Event that the system state $\boldsymbol{x}$ is less than or equal to the i-th maximum lower vector $\boldsymbol{c}^{(i)}$
$\boldsymbol{r} = (r_1, \cdots, r_n)$	Minimal upper vector, an extension of the minimal path-set tailored for coherent multi-state-component systems
γ-MP	γ-minimal path, which is referred to as the minimal upper vector in two-terminal maximum flow analysis
N_r	Number of all minimal upper vectors
A_i^*	Event that the system state $\boldsymbol{x}$ is larger than or equal to the i-th minimal upper vector $\boldsymbol{r}^{(i)}$
$\mathcal{U}^{(l)}$	Unspecified domain for decomposition at the l-th level of the state space decomposition method
box $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$	Hyperrectangular domain with minimal state $\boldsymbol{\alpha} \triangleq (\alpha_1, \cdots, \alpha_n)$ and maximal state $\boldsymbol{\beta} \triangleq (\beta_1, \cdots, \beta_n)$
$\boldsymbol{x}^o, \boldsymbol{x}^*$	Critical state vectors used for decomposing $\mathcal{U}^{(l)}$
$\underline{x_d^o}$ (resp. $\underline{x_d^*}$)	largest state in $\boldsymbol{s}_d = (s_{d,1}, \cdots, s_{d,n_d})$ that is smaller than x_d^o (resp. x_d^*)
$\mathcal{F}$	Feasible domain resulting from the decomposition of $\mathcal{U}^{(l)}$
$\mathcal{I} = \bigcup_{j=1}^n \mathcal{I}_j$	Infeasible domain resulting from the decomposition of $\mathcal{U}^{(l)}$ (The set $\mathcal{I}$ can be partitioned by n hyperrectangles $\mathcal{I}_j, j = 1, \cdots, n$)
$\mathcal{U} = \bigcup_{j=1}^n \mathcal{U}_j$	Unspecified domain resulting from the decomposition of $\mathcal{U}^{(l)}$ (The set $\mathcal{U}$ can be partitioned by n hyperrectangles $\mathcal{U}_j, j = 1, \cdots, n$)
$\mathcal{G}^{(l)}$	Graph for decomposition at the l-th level of the recursive decomposition method
$\boldsymbol{a} = (a_1, \cdots, a_n)$	Index set of a minimal path in $\mathcal{G}^{(l)}$
$\mathcal{G}_0^{(l)}, \cdots, \mathcal{G}_k^{(l)}$	Subgraphs derived from decomposing $\mathcal{G}^{(l)}$ with the minimal path $\boldsymbol{a}$
N	Sample size of sampling-based-methods
$\widehat{p}^{(MCS)}$	Crude Monte Carlo estimator
m	Number of nodes in the network
$\lvert \cdot \rvert$	Cardinality (Number of elements) of a set
$\boldsymbol{z} = (z_1, \cdots, z_m)$	Auxiliary Boolean variables in counting-based method, representing the color assigned to each network node
T_d	Artificial repair time of each d-th edge
$\mathcal{E}(t)$	Edge set that collects all functional edges at artificial time t
$\mathcal{G}(t) = \mathcal{G}(\mathcal{V}, \mathcal{E}(t))$	Network snapshot at artificial time t
T^*	Artificial critical time that the network first becomes functional in creation-process-based methods
$\Pi = (\Pi_1, \cdots, \Pi_n)$	The order in which the edges are repaired with Π_1 denoting the index of the first repaired edge

$[\mathbf{\Pi}]$	Critical order such that $\mathcal{G}(\mathcal{V}, \{e_{\Pi_1}, \cdots, e_{\Pi_{[\mathbf{\Pi}]}}\})$ is functional and $\mathcal{G}(\mathcal{V}, \{e_{\Pi_1}, \cdots, e_{\Pi_{[\mathbf{\Pi}]-1}}\})$ is not
$\Delta T_0 \triangleq T_{\Pi_1}$	Sojourn time of $\mathcal{G}(t)$ in $\emptyset$
$\Delta T_l \triangleq T_{\Pi_{l+1}} - T_{\Pi_l}$	Sojourn time of $\mathcal{G}(t)$ in $\mathcal{G}(\mathcal{V}, \{e_{\Pi_1}, \cdots, e_{\Pi_l}\})$, where $l = 1, \cdots, n-1$
$\widehat{p}_f^{(pMCS)}$	Permutation Monte Carlo estimator
$T_{d,i}$	Deadline for restoring the d-th edge, X_d, to at least its i-th state $s_{d,i}$
M	Number of levels in multi-level splitting methods
$F_0 \subset F_1 \cdots \subset F_M = F$	Intermediate failure domains in multi-level splitting methods
$\widehat{p}_f^{(MS)}$	Multi-level splitting estimator
$-\inf = \gamma_0 < \gamma_1 < \cdots < \gamma_M = \gamma$	Intermediate levels in multi-level splitting methods
N_l	Sample size at the l-th level of multi-level splitting methods
R_l	Number of seeds that are generated at l-th level and are located in F_{l+1}
$n_1^{(k)}$	Number of replications of the k-th seed at level l
η_l	Splitting factor in fixed splitting
p_0	Conditional failure probability used for adaptively selecting the intermediate threshold in multi-level splitting methods
$\widehat{p}_f^{(IS)}$	Importance sampling estimator
$p_{IS}(\boldsymbol{x})$	Importance sampling distribution
$p_{IS}^*(\boldsymbol{x})$	Optimal importance sampling distribution
$\boldsymbol{c}^E$ (resp. $\boldsymbol{c}^F$)	Event vector that indicates the membership of each system state $\boldsymbol{x}$ in $\Omega_{\boldsymbol{X}}$ to the component event E (resp. the failure event F)
$\boldsymbol{p}^{\Omega_{\boldsymbol{X}}}$	Probability vector comprising the probability of each system state $\boldsymbol{x}$ in $\Omega_{\boldsymbol{X}}$
$\mathbf{\Phi} = (\Phi(0), \cdots, \Phi(n))$	System signatures, which denote the conditional reliability given exactly $0, \cdots, n$ functional components

Chapter 1

Introduction

1.1 Background and scope

Infrastructure networks, including road networks, water supply systems, underground pipelines, and power grids, constitute the backbone of modern society, facilitating the delivery of products and services that underpin our daily lives. However, these systems are susceptible to various threats, ranging from natural disasters and physical attacks to aging and deterioration.

Fig. 1.1 illustrates the annual frequency of significant blackouts in the U.S. over the last decade [51]. These failures resulted in severe consequences. For instance, in February 2021, the state of Texas in the U.S. was hit by three severe winter storms, resulting in state-wide power failures and subsequent shortages of water, food, medical service, and heating. This crisis claimed at least 111 people's lives, and the property damage reached an estimated minimum of $155 billion, equivalent to nearly 7.4% of Texas's GDP that year [22].

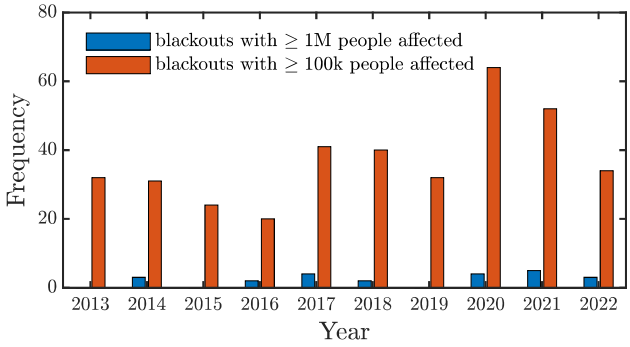


Figure 1.1: The frequency of significant blackouts in the U.S. (2013-2022)

It is, therefore, important and necessary to efficiently manage the safety of infrastructure networks throughout their design and operational lifespans. One primary challenge in this regard is to appropriately quantify the uncertainty inherent in both the network's structure and its operating environment and to accurately assess the network's reliability or, conversely, its probability of failure. The

definition of network failure may vary significantly depending on the specific context, objectives, and requirements of the network. Also, different utility companies, stakeholders, and regulatory bodies establish their own distinct standards and criteria. Often, network failure can be defined as the event of a performance metric surpassing or falling below a specified threshold, e.g., the blackout size of a power network exceeding 5%. In practical applications, the network performance metric is seldom a deterministic value; rather, it is frequently modeled as a random variable to reflect its inherent randomness.

The classic uncertainty quantification framework has two main ingredients: (1) a deterministic performance function $g(\cdot)$ that maps a set of n input variables $\boldsymbol{X} = (X_1, \cdots, X_n)$, which can include hazard parameters, site conditions, material properties, to the network performance metric $g(\boldsymbol{X})$; and (2) a probabilistic model $p_{\boldsymbol{X}}(\boldsymbol{x})$ that characterizes the uncertainty of the input variables. The notation $\boldsymbol{x}$ represents a specific realization of $\boldsymbol{X}$ that takes values from the set $\Omega_{\boldsymbol{X}}$. The failure probability of the network, denoted as p_f, can then be expressed as the probability that the performance metric $g(\boldsymbol{X})$ exceeds or falls below a specified threshold γ, i.e.,

$$p_f = \int_F p_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \mathbb{E}_{p_{\boldsymbol{X}}} \left[\mathbb{I}\{\boldsymbol{X} \in F\} \right], \tag{1.1}$$

where F denotes the failure domain that is characterized by $g(\cdot)$ and γ, and $\mathbb{I}\{\cdot\}$ represents the indicator function. This reliability calculation problem forms the basis of network optimization, design, resilience planning, maintenance strategy, among others, so it plays a central role in ensuring the safety of critical infrastructures.

While it may appear straightforward, the computation of p_f is far from trivial. Even for basic performance metrics like connectivity and maximum flow between a pair of nodes, the calculation of p_f becomes NP-hard [11], i.e., no efficient exact algorithms that run in polynomial time in the input dimension is likely to exist. Note that there are more refined and informative complexity classes for these problems, as referenced in Provan and Ball [116], however, we will not delve into that level of detail within this thesis. Indeed, the infrastructure networks often consist of hundreds of components, and it is thus more pragmatic to turn to approximation and sampling techniques rather than exact algorithms. For physics-driven performance metrics, the respective p_f calculation problems are even more challenging. Determining these metrics requires running complex and computationally intensive simulation models. Moreover, these models often do not posses a specified structure such as monotonicity, also known as coherency [119], that can be exploited to devise efficient algorithms. Additionally, critical infrastructures are designed to be highly reliable, so their failure probability is notably small. Thus, Eq. (1.1) characterizes a rare event estimation problem.

The above challenges favor the use of sampling-based methods due to the generality inherent to these approaches. There is a substantial body of literature dedicated to network reliability assessment using crude Monte Carlo simulation (MCS) [142, 92]. However, crude MCS is, in general, impractical for estimating rare events, as the number of required samples, along with the associated calls to the potentially expensive performance function required to obtain an accurate estimate, scales with $O\left(\frac{1}{p_f}\right)$. Additionally, a variety of variance reduction techniques have been developed for addressing structural reliability problems, such as subset simulation (SuS) [8] and cross entropy (CE) method [123], but their applicability to network reliability cannot be taken for granted. This is because network performance functions $g(\boldsymbol{x})$ are not necessarily continuous and frequently involve binary and multi-state inputs. Such discontinuity can pose additional challenges to network reliability assessment. This problem can be circumvented by converting discrete network performance into

a continuous one [53, 38]. However, such transformations are typically tailored to the specific problem at hand or necessitate numerous calculations of the original network performance, leading to significantly increased costs.

In summary, the reliability calculation problem in Eq. (1.1) is central to ensuring the safety of our critical infrastructures, yet it is quite challenging to solve. While sampling-based methods offer a promising approach to tackle this issue, there are several challenges in the application of existing methods to the estimation of rare failure events in networks, which motivates the research conducted in this thesis.

1.2 Research objectives

The primary objective of this thesis is to investigate and develop efficient sampling-based methods for estimating rare failure events within infrastructure networks. The significance, necessity, and justification behind this objective have already been elaborated in the background section. In this section, we outline some of the specific goals.

- Offering a comprehensive overview of the state-of-the-art network reliability techniques while also highlighting their strengths and weaknesses.
- Substantiating and expanding the rare event estimation techniques originally introduced for addressing structural reliability issues to the network reliability assessment.
- Identifying critical network components as a by-product of the sampling-based algorithms.
- Establishing unified benchmarks to facilitate the comparison of various network reliability algorithms and exploring potential combinations between them.

1.3 Contributions

We next outline the main contributions of this thesis, which are based on four journal articles, each comprising a chapter in Part II. In Chapters 4-6, we expand upon two widely adopted variance reduction techniques, namely the standard SuS and CE, to assess network reliability and enhance their performance in the context of physics-driven network performance metrics. Subsequently, the strengths and weaknesses of the proposed methods are exemplified in Chapter 7 through unified benchmarks.

Adaptive effort subset simulation algorithms

Chapter 4 is based on the original publication [30], whose main contributions are:

- *Conceptual*: Identify the discontinuity issue in network performance metrics and illustrate its influence on standard SuS.

- *Algorithmic*: Propose the adaptive effort subset simulation (aE-SuS) algorithm that modifies the adaptive selection of the intermediate domains.
- *Algorithmic*: Introduce a novel independent Metropolis-Hastings algorithm for efficient sampling in discrete space.

Bayesian improved cross entropy method

Chapter 5 is based on the original publication [31], whose main contributions are:

- *Theoretical*: Prove that under perfect sampling assumptions, the adaptation of the intermediate target distributions in the improved cross entropy (iCE) method is unique and convergent.
- *Theoretical*: Prove that for fitting an exponential parametric family, optimizing the CE is equivalent to applying self-normalized importance sampling. This justifies the use of the effective sample size in the iCE method.
- *Conceptual*: Identify the overfitting issue in CE-based importance sampling with the categorical distribution.
- *Methodological*: Incorporate the Bayesian statistics into the CE-based importance sampling to mitigate the overfitting issue.
- *Algorithmic*: Introduce the Bayesian improved cross entropy (BiCE) method for rare event estimation in networks, resulting in an unbiased estimator.

Bayesian improved cross entropy method with categorical mixture models

Chapter 6 is based on the original publication [32], whose main contributions are:

- *Methodological*: Introduce the BiCE method with categorical mixture models that enables tackling network reliability problems with complex failure domains.
- *Algorithmic*: Propose a generalized expectation-maximization algorithm for approximating the maximum-a-posteriori estimate of categorical mixture models with weighted data.
- *Methodological*: Use the failure samples from the BiCE method to compute the component importance measure.

Adaptive Monte Carlo methods for estimating rare events in power grids

Chapter 7 is based on the original publication [33], whose main contributions are:

- *Empirical*: Establish unified power flow benchmarks and propose the use of a new efficiency metric for assessing the performance of various adaptive Monte Carlo simulation (adaptMCS) methods.

- *Algorithmic*: Propose a hybrid approach that combines the strengths of both aE-SuS and annealed particle integration methods.
- *Algorithmic*: Compute the component importance measure as a function of the threshold γ using one single run of the aE-SuS method.

1.4 Outline

This thesis consists of two parts. Part I provides a comprehensive overview of the state-of-the-art network reliability techniques. We begin in Section 2.1 by introducing the primary performance metrics central to this thesis, followed by an overview of non-sampling-based methods in Section 2.2. These methods include binary decision diagram, various cut(path)-based techniques, and state space decomposition methods. We delve into their application to both binary and multi-state inputs while also highlighting their limitations in the context of infrastructure network analysis. Section 2.3 is dedicated to existing sampling-based methods for network reliability assessment. In particular, we detail the creation process whereby the discrete network performance metric is transformed into a continuous one, and summarize various variance reduction techniques that can be subsequently incorporated for estimating rare events in networks, among which we place particular emphasis on multi-level splitting and CE-based importance sampling. An overview of other widely adopted network reliability techniques is also provided in this section.

Part II consists of the main research conducted in this thesis. Chapter 4 extends a variant of the adaptive multi-level splitting (AMS) method, known as the SuS in the context of civil engineering, to accommodate discrete network performance. Chapters 5 and 6 are dedicated to CE-based methods, where we identify and address the issue of overfitting by proposing a Bayesian estimator. The adaptation strategy of the iCE method is also substantiated. In Chapter 7, we establish unified power flow benchmarks and introduce a novel efficiency metric for evaluating and comparing the performance of different sampling-based methods. Additionally, the calculation of component importance measures is discussed as a by-product of the proposed sampling-based methods, with further details provided in Chapters 6 and 7.

CHAPTER 2

A review of the state-of-the-art network reliability methods

2.1 Network performance metrics

We first present the key network performance metrics central to this thesis: connectivity, maximum flow, and physics-driven metrics such as the blackout size of a power grid, along with their respective reliability evaluation problems.

2.1.1 Connectivity

A network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$ is connected if all terminal nodes indexed in $\mathcal{K}$ are connected. This metric is widely used for characterizing the performance of communication networks and circuit-switched networks, among various other applications [20]. The corresponding network reliability is defined as the probability that the network is connected under random edge or node failure. In principle, each undirected edge can be substituted with two directed edges pointing in opposite directions, and for a directed graph, it is sufficient to only consider the edge failure [12]. Hence, in connectivity-based reliability problems, it is often assumed that the nodes are perfect, but each d-th edge can either fail or not fail with probability p_d and $1 - p_d$, respectively. The states of the edges are often assumed to be independent if no dependence structure is specified.

The reliability calculation problem can be stated as follows. Let the vector $\boldsymbol{X} = (X_1, \cdots, X_n)$ collect the state of all n edges, whose probability mass function (PMF) is denoted as $p_{\boldsymbol{X}}(\boldsymbol{x})$. The distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$ follows an independent Bernoulli distribution. Given the edge state vector $\boldsymbol{x}$, the connectivity of the network $\mathcal{G}$, referred to as $g^{(\text{conn})}(\boldsymbol{x})$, can be determined through numerous searching algorithms, e.g., the breath-first-searching algorithm [108]. We adopt the convention that the performance metric $g^{(\text{conn})}(\boldsymbol{x})$ equals one if the network is connected and zero otherwise. The network reliability can then be written as $\Pr(g^{(\text{conn})}(\boldsymbol{X}) = 1)$, and by contrast, the failure probability, denoted as p_f, equals $\Pr(g^{(\text{conn})}(\boldsymbol{X}) = 0) = 1 - \Pr(g^{(\text{conn})}(\boldsymbol{X}) = 1)$. Note that $g^{(\text{conn})}$ is also known as the structure function.

Depending on the number of the terminal nodes, i.e., $|\mathcal{K}|$, one distinguishes two-terminal connectivity with $|\mathcal{K}| = 2$, all-terminal connectivity with $|\mathcal{K}| = m$, and K-terminal connectivity with $|\mathcal{K}| = K$. Note that in contrast to the notation convention in graph theory, where m is the number of edges and n is the number of nodes, we use n to denote the number of edges and m for the nodes. This aligns with the tradition in rare event estimation, where n signifies the dimension of the problem, which in most cases corresponds to the number of edges. The respective reliability calculations of all these three cases are NP-hard [11], suggesting that no efficient algorithms is likely to exist for exact inference.

2.1.2 Maximum flow

In many applications, such as transportation networks, road networks, and power transmission networks, the network performance cannot be adequately characterized solely by the connectivity among a set of nodes. Instead, it is intricately linked to attributes like capacity, load, length, or traveling time of the network components [96]. To analyze the reliability of these systems, a more informative network performance and a more flexible input distribution are therefore needed.

One such performance metric is the maximum flow between a pair of nodes. This metric can be effectively evaluated using the augmenting path algorithm given that all edge capacities are integer [108]. Additionally, in the next subsection, we will delve into another metric that assesses the blackout size of a power network. This metric is driven by physics and operation strategies and necessitates more complex simulation models.

To align with Subsection 2.1.1, we let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ denote a network, with node set $\mathcal{V}$ and edge set $\mathcal{E}$, but now the input vector $\boldsymbol{X} = (X_1, \cdots, X_n)$ characterizes the flow capacity of each edge. In particular, each d-th edge independently takes the capacity $0 = s_{d,1} < \cdots < s_{d,n_d}$ with probability $p_{d,1}, \cdots, p_{d,n_d}$, respectively, where n_d denotes the number of possible states of the d-th edge. The input vector $\boldsymbol{X}$ therefore follows an independent categorical distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$. Given the current capacity configuration $\boldsymbol{x}$, the performance function $g^{(\mathrm{mf})}(\boldsymbol{x})$ calculates the maximum flow that can be delivered from the source to the sink, and the failure probability p_f is then defined as the probability the maximum flow $g^{(\mathrm{mf})}(\boldsymbol{X})$ is less than a specified flow demand γ, i.e., $p_f = \Pr(g^{(\mathrm{mf})}(\boldsymbol{X}) < \gamma)$. Also, since $\boldsymbol{X}$ is discrete and takes finite number of states, the network performance metric $g^{(\mathrm{mf})}(\boldsymbol{X})$ is also a multi-state random variable. This is also true for physics-driven metrics.

The maximum flow problem can be interpreted as an extension of the two-terminal connectivity-based problems, and hence, the respective reliability calculation must also be NP-hard.

Both connectivity-based and maximum flow problems are widely investigated in operations research. Due to the frequent absence of a specific dependence structure among component states, these studies mainly focus on independent inputs. While this assumption is unrealistic for numerous practical applications, the methodologies developed for connectivity or maximum flow problems can often be adapted to dependent cases after modifications. In fact, the identification of failure modes and the calculation of reliability are usually decoupled in numerous non-sampling methods, and sampling-based methods are directly applicable if it is feasible to sample from the input distribution and to calculate the probability of a single sample. These will be briefly discussed in Section 2.5.

2.1.3 The blackout size within a power network

In power transmission networks, the flow of electrical power is subject to Kirchhoff's law and is influenced by various operational strategies. We focus on two primary types of power flow analyses in this thesis: the standard power flow analysis and the optimal power flow analysis [141]. The standard power flow analysis considers the possibility of cascading failures, where an initial component failure can trigger subsequent failures due to load redistribution. By contrast, the optimal power flow analysis aims to prevent such cascading failures by implementing optimal operation strategies that minimize a predefined cost function. In both cases, it is possible that the power demands cannot be fully met, potentially resulting in localized blackouts. The extent to which the original load cannot be satisfied serves as a quantifiable measure of the blackout size and can be used as a performance metric for assessing the reliability of power grids.

The respective reliability calculation problem can be formally stated as follows. The input vector $\boldsymbol{X} = (X_1, \cdots, X_d)$ now denotes the states of network components such as transmission lines, power generators, and connecting buses, following a joint categorical distribution. The performance function, denoted as $g^{(\mathrm{bz})}(\boldsymbol{x})$, outputs the blackout size given the current configuration (or component states) $\boldsymbol{x}$. The system reliability is then defined as the probability that $g^{(\mathrm{bz})}(\boldsymbol{X})$ is less than a specified threshold γ, and conversely, the failure probability p_f equals $\Pr(g^{(\mathrm{bz})}(\boldsymbol{X}) \geq \gamma)$.

It's important to note that, unlike metrics such as connectivity and maximum flow, the blackout size of a network is not necessarily coherent; in other words, the system's performance function, $g^{(\mathrm{bz})}(\boldsymbol{x})$, is not always monotonic in $\boldsymbol{x}$. In some instances, improving the capacity of a generator can lead to increased load redistribution and higher potential for cascading failures in other branches, ultimately degrading system performance (resulting in a larger blackout size). Additionally, power flow analysis often involves intricate and potentially expensive simulation models, which can pose additional challenges when conducting reliability analyses.

A brief summary of the reliability calculation problems linked to the aforementioned network performance metrics is provided in Table 2.1. For all these metrics, the exact calculation of the reliability is NP-hard, and in most cases, one has to resort to different approximation techniques, either deterministic or stochastic.

Table 2.1: Network performance metrics and their reliability calculation problems

	connectivity, $g^{(\mathrm{conn})}(\cdot)$	maximum flow, $g^{(\mathrm{mf})}(\cdot)$	blackout size, $g^{(\mathrm{bz})}(\cdot)$
metric's type	binary	multi-state	multi-state
calculation method	breadth(depth)-first-search	augmenting path	power flow analysis
input vector, $\boldsymbol{X}$	edge state	edge capacity	component state
input distribution, $p_{\boldsymbol{X}}$	independent Bernoulli	independent categorical	categorical
failure criterion	not all terminal nodes are connected	the maximum flow is below the threshold γ	the blackout size surpasses the threshold γ

2.2 Enumeration and approximation techniques

In this section, we present three commonly utilized non-sampling methods: the binary decision diagram (BDD), cut(path)-based methods, and state space decomposition (SSD) methods. These methods provide efficient and exact failure probability calculation in certain scenarios. Alternatively, they can be halted prematurely to provide a failure probability bound, thereby transforming into deterministic approximation techniques. A detailed comparison among the three methods is presented in Section 2.5.

2.2.1 Binary decision diagram (BDD) and its applications

2.2.1.1 Binary decision diagram (BDD)

The BDD was first proposed by Lee [88] for representing and designing switching circuits. Therein, the diagram is a composition of 'instructions', each taking the form of an if-then-else function $ite(x, y, z)$ that directs to variable y if $x = 1$ (x is true) and to variable z otherwise and continues proceeding with the if-then-else functions in y and z. The BDD has been subsequently employed as a powerful tool for representing and manipulating general Boolean functions [3]. As an example, Fig. 2.1 depicts the BDD for the Boolean function $f(x_1, x_2, x_3) = (\neg x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$, where $\neg, \vee$, and $\wedge$ denote the basic Boolean operations: logic NOT, logic OR, and logic AND, respectively.

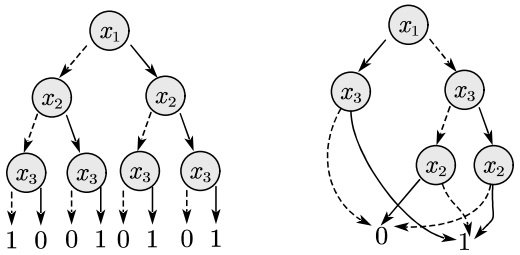


Figure 2.1: The BDD (left) and the ROBDD (right) of the Boolean function $f(x_1, x_2, x_3)$. Each node in the two diagrams corresponds to a Boolean variable x_i and has two outgoing branches, of which the solid one represents setting x_i to 1, and the dashed one represents $x_i = 0$. Any path from the root to the leaf 1 in the two diagrams represents an assignment of the Boolean variables x_1, x_2, x_3 that gives $f(x_1, x_2, x_3) = 1$. The ROBDD shown on the right-hand side of the figure is constructed with the order $x_1 \to x_3 \to x_2$.

Nowadays, the terminology BDD almost always refers to the reduced and ordered binary decision diagram (ROBDD), also known as the functional graph [21]. Compared to the BDD, the ROBDD relies on a specified order of input decision variables and contains no redundant nodes and duplicate subgraphs. For instance, the ROBDD of the Boolean function $f(x_1, x_2, x_3)$ with order $x_1 \to x_3 \to x_2$ is illustrated in Fig. 2.1, which is essentially a composition of ite functions as follows:

$$f(x_1, x_2, x_3) = ite\left(x_1, ite(x_3, 1, 0), ite\left(x_3, ite(x_2, 1, 0), ite(x_2, 0, 1)\right)\right).$$

One can prove that a Boolean function's ROBDD is unique (up to isomorphism), and any other BDD representation of the same function will have an equal or larger number of nodes than the ROBDD [21].

More importantly, as a knowledge compilation strategy, operations on the Boolean function, such as $\vee$, $\wedge$, $\neg$, can be directly applied to its ROBDD through tailored algorithms without decompression. The complexity of these algorithms directly relates to the size (or the number of nodes) of the ROBDD and further depends on the specified order of the decision variables. Unfortunately, finding the optimal order that minimizes a ROBDD's size is a co-NP-complete problem [21], and in practice, the order is chosen according to heuristics [137].

Given the Boolean function's expression, e.g., the logic gate network's structure, the associated ROBDD can be constructed iteratively in a bottom-up manner, as outlined below [137]:

1. Represent each Boolean decision variable X_i as a binary decision tree, that is $ite(x_i, 1, 0)$, and recursively assemble them into blocks and the Boolean function according to the following manipulation rule:

$$\begin{aligned} G \lozenge H &= ite(x, G_{x=1}, G_{x=0}) \lozenge ite(y, H_{y=1}, H_{y=0}) \\ &= \begin{cases} ite(x, G_{x=1} \lozenge H_{x=1}, G_{x=0} \lozenge H_{x=0}), & \text{index}(x) = \text{index}(y) \\ ite(x, G_{x=1} \lozenge H, G_{x=0} \lozenge H), & \text{index}(x) < \text{index}(y) \\ ite(y, G \lozenge H_{x=1}, G \lozenge H_{x=0}), & \text{index}(x) > \text{index}(y) \end{cases}. \end{aligned} \tag{2.1}$$

   Here, G and H are two Boolean functions, and x and y denote Boolean decision variables, with orders specified by index(x) and index(y), respectively. $G_{x=1}$ and $G_{x=0}$ are subfunctions of G conditional on $x = 1$ and $x = 0$, respectively. The symbol $\lozenge$ represents a logic operation, either $\wedge$ or $\vee$.

2. Reduce the resulting BDD to ROBDD by the *reduction* algorithm described in Bryant [21].

As an example, for building up the ROBDD of $f(x_1, x_2, x_3)$, one first constructs the BDD for $\neg x_1, \neg x_2, \neg x_3, x_1, x_2$, and x_3. These BDDs are then assembled into $(\neg x_1 \wedge \neg x_2 \wedge \neg x_3)$, $x_1 \wedge x_3$, and $x_2 \wedge x_3$, and finally into the function $f(x_1, x_2, x_3)$. The 'reduction' algorithm is subsequently invoked to get the final ROBDD of f. We refer the readers to [21] for more implementation details. Alternatively, if the Boolean function's expression is only provided implicitly, e.g., in connectivity-based network reliability assessment, the construction of ROBDD follows a top-down manner [72, 68]. This will be detailed in the next subsection.

2.2.1.2 Reduced and ordered binary decision diagram in reliability analysis

One of the major advantages of using ROBDDs for representing the performance function in reliability analysis is that any path from the root to the leaf in the ROBDD represents a disjoint event. Consequently, the probability of the function being 1 (or true) can be computed by adding up the probability of the path from the root to leaf 1, and this can be done through a recursive algorithm with complexity linear to the ROBDD's size, i.e., the number of nodes in the ROBDD. Note that

this algorithm is not necessarily efficient since the ROBDD's size can grow exponentially with the number of variables, especially when an improper order is chosen. In the following, we introduce two main applications of ROBDDs in reliability analysis: fault tree analysis and connectivity-based network reliability (i.e., the network performance is measured by connectivity).

As a logic gate network, the fault tree can be naturally represented as an ROBDD, based on which Rauzy devised algorithms that allow for efficient computation of both the minimal cut and the probability of the root event [120]. This is widely acknowledged as the first application of the ROBDD in reliability analysis. For multi-state systems with multi-state components, the system structure can be modeled by a group of fault trees, in which each root indicates whether the system is in a specific state, so n_{sys} fault trees are required for modeling an n_{sys}-state system. Additionally, each leaf, i.e., the basic event, is linked to a component state, and an n_d-state component is represented by n_d leaves. Since a component can only take one state at a time, the basic events are no longer independent, and a modified construction procedure of the ROBDD was proposed to account for such restrictions [140]. The probability of the root event can be efficiently calculated through a modified recursive algorithm. Note that it is the size (the number of nodes) of the ROBDD rather than the number of basic events that really matters since the former directly relates to the efficiency of computing the root event's probability. Shrestha and Xing [128] find that the ROBDD's size is usually reduced by using a logarithmic encoding of the multi-state components, where the n_d component states are encoded with $\lceil \log_2 n_d \rceil$ auxiliary Boolean variables. Besides the ROBDD, the multi-state multi-valued decision diagram (MMDD) is another promising method for multi-state fault tree analysis. The MMDD is a natural extension of the ROBDD for encoding functions with multi-state input and a Boolean output [138]. In the MMDD, each node can have multiple descendants. Algebraic decision diagram is another extension of the ROBDD [9], where the leaves may take multiple values different from 0 and 1.

For all-terminal connectivity-based problems, Imai et al. [72] constructed the ROBDD by iteratively performing the following two steps: (1) network decomposition using edge construction/deletion (one edge at a time) and (2) merging equivalent subgraphs. The undetermined subgraphs with the same partition of elimination front, or boundary components, are proven to be equivalent and, hence, are merged at each iteration. This idea was then extended to K-terminal cases [68]. The algorithm's complexity grows exponentially with the maximum number of boundary components in all iterations, which is further related to the ordering of edges [68]. Hardy et al. [68] suggested a breadth-first-search for ordering edges. In addition, dependent component failures can also be addressed by ROBDD-based methods [137].

2.2.2 Cut(path)-based network reliability assessment

2.2.2.1 Minimal cut(path) and their extensions

In networks with binary components, a minimal (or irreducible) cut is a group of components whose removal from the network results in network failure, and no proper subset of it is sufficient to cause network failure [12]. This concept can be extended to coherent multi-state-component systems, in which each minimal cut represents a maximum lower vector, denoted as $\boldsymbol{c} = (c_1, \cdots, c_n)$ [74]. Here, n is the number of network components (or dimension of the problem). Any state $\boldsymbol{x} = (x_1, \cdots, x_n)$

that is greater than $\boldsymbol{c}$, i.e., $x_d \geq c_d$ for each $d = 1, \cdots, n$, and $\boldsymbol{x} \neq \boldsymbol{c}$, falls within the safe domain, indicating the vector is maximal. By contrast, due to the coherency of the system, any state that is smaller than or equal to $\boldsymbol{c}$, i.e., $x_d \leq c_d$ for each $d = 1, \cdots, n$, leads to network failure. This maximum lower vector is referred to as the γ-minimal cut (γ-MC) in two-terminal maximum flow analysis, where one estimates the probability that the maximum flow from the source to the sink is less than γ [75]. The minimal cut and the γ-MC of a toy network example are illustrated in Fig. 2.2.

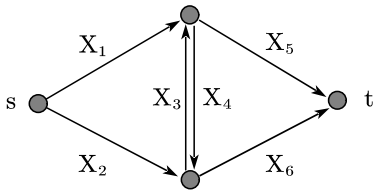


Figure 2.2: A toy network and its minimal cuts. s and t denote the source node and the sink node, respectively. For connectivity-based problems, X_i indicates whether the i-th edge, e_i, is connected. The corresponding minimal cuts can be enumerated as: (e_1, e_2), (e_5, e_6), (e_1, e_3, e_6), and (e_2, e_4, e_5). For maximum-flow-based problems, X_i represents the multi-state capacity of the i-th edge, e_i. Assume that the threshold γ is 2, and each edge's capacity is restricted to $\{0, 3, 5\}$ unit. Then $\boldsymbol{c} = (0, 0, 5, 5, 5, 5)$ is a maximum lower vector (or a γ-MC).

To illustrate the importance of this concept in network reliability assessment, let A_i denote the event that the system state $\boldsymbol{x} = (x_1, \cdots, x_n)$ is less or equal to the i-th γ-MC, $\boldsymbol{c}^{(i)}$, i.e., $A_i \triangleq \{\boldsymbol{x} \mid \boldsymbol{x} \leq \boldsymbol{c}^{(i)}\}$, $i = 1, ..., N_c$, where N_c is the number of all γ-MCs. Efficiently searching for all γ-MCs poses a significant challenge [75]. Even for connectivity-based problems, counting the minimal cuts with the smallest cardinality cannot be accomplished in polynomial time, i.e., it is intractable [11]. The state-of-the-art algorithms for identifying all γ-MCs are only practical for small networks, with a few tens of components [109]. The system failure event F then becomes the union of all A_i's, and the failure probability can be expressed as follows:

$$\Pr(F) = \Pr\left(\bigcup_{i=1}^{N_c} A_i\right). \tag{2.2}$$

If only a subset of all γ-MCs is discovered, Eq. (2.2) provides a lower bound for $\Pr(F)$.

Alternatively, one can define the minimal upper vector, denoted as $\boldsymbol{r}$, where any state $\boldsymbol{x}$ that is strictly smaller than r leads to network failure, and any state $\boldsymbol{x}$ that is greater than or equal to r ensures the system's functionality. In two-terminal maximum flow analysis, the vector $\boldsymbol{r}$ corresponds to the γ-minimal path (γ-MP) [95]. Let N_r denote the number of all γ-MPs, and define $A_i^* \triangleq \{\boldsymbol{x} \mid \boldsymbol{x} \geq \boldsymbol{r}^{(i)}\}$. The system functional event S can be written as the union of all A_i^*'s:

$$\Pr(S) = \Pr\left(\bigcup_{i=1}^{N_r} A_i^*\right). \tag{2.3}$$

Both Eq. (2.2) and Eq. (2.3) require counting the elements within a union of sets, and various tools originally developed in set theory can be leveraged for this purpose. These include the principle of inclusion and exclusion and the sum of disjoint products [34].

2.2.2.2 Cut(path)-based network reliability assessment

Based on the principle of inclusion and exclusion, the failure probability $\Pr(F)$ can be decomposed as follows:

$$\Pr(F) = \Pr\left(\bigcup_{i=1}^{N_c} A_i\right) = \sum_{1\le i\le N_c} \Pr(A_i) - \sum_{1\le i<j\le N_c} \Pr(A_i \cap A_j) + \cdots + (-1)^{N_c-1} \Pr(A_i \cap \cdots \cap A_{N_c}). \tag{2.4}$$

In many network reliability problems, the components' states are assumed to be statistically independent, so for any subset $\mathcal{S}$ of $\{1, \cdots, N_c\}$, $\Pr(\cap_{\mathcal{S}} A_i)$ can be easily calculated as

$$\Pr(\cap_{\mathcal{S}} A_i) = \prod_{d=1}^{n} \Pr\left(X_d \le \min_{i\in\mathcal{S}} c_d^{(i)}\right) = \prod_{d=1}^{n} \min_{i\in\mathcal{S}} \Pr\left(X_d \le c_d^{(i)}\right). \tag{2.5}$$

However, the number of terms in Eq. (2.4) equals $2^{N_c} - 1$. Considering that the number of minimal cuts N_c can also grow exponentially with the number of network components n [12], the complexity of determining the failure probability with Eq. (2.4) in such cases is therefore doubly exponential in n, which is only feasible for networks with limited size.

Another frequently employed combinatorial counting technique is the sum-of-disjoint-products, which decomposes the failure probability $\Pr(F)$ as follows:

$$\begin{aligned}
\Pr(F) &= \Pr\left(\bigcup_{i=1}^{N_c} A_i\right) = \Pr(A_1) + \Pr(\overline{A_1} A_2) + \cdots + \Pr\left(\left(\overline{\cup_{i=1}^{N_c-1} A_i}\right) A_{N_c}\right) \\
&= \underbrace{\Pr(A_1)}_{\text{1-st term}} + \underbrace{\Pr(A_2) - \Pr(A_1A_2)}_{\text{2-nd term}} + \cdots + \underbrace{\Pr(A_{N_c}) - \Pr\left(\left(\cup_{i=1}^{N_c-1} A_i\right) A_{N_c}\right)}_{\text{Nc-th term}} \\
&= \underbrace{\Pr(A_1)}_{\text{1-st term}} + \underbrace{\Pr(A_2) - \Pr(A_1A_2)}_{\text{2-nd term}} + \cdots + \underbrace{\Pr(A_{N_c}) - \Pr\left(\cup_{i=1}^{N_c-1} (A_iA_{N_c})\right)}_{\text{Nc-th term}}.
\end{aligned} \tag{2.6}$$

Expanding each $\Pr\left(\cup_{i=1}^{j-1} (A_iA_j)\right)$ in Eq. (2.6) with the principle of inclusion and exclusion, however, yields no extra benefit, since the total number of terms remains $2^{N_c}-1$, the same as that in Eq. (2.4). To calculate $\Pr\left(\cup_{i=1}^{j-1} (A_iA_j)\right)$ more efficiently, Zuo et al. [145] introduced a recursive subroutine, denoted as $\mathrm{PrU}(B_1, \cdots, B_N)$, which computes the probability of the union of events $B_1, \cdots, B_N$ using the following recursion formula:

$$\mathrm{PrU}(B_1, \cdots, B_N) = \Pr(B_N) + \mathrm{PrU}(B_1, \cdots, B_{N-1}) - \mathrm{PrU}(B_1B_N, \cdots, B_{N-1}B_N). \tag{2.7}$$

The input events $B_1, \cdots, B_N$ are checked each time when calling $\mathrm{PrU}(\cdot)$, and if $B_i \subseteq B_j$ holds for any $i \ne j$ in $1, \cdots, n$, B_i can be safely removed since $B_i \cup B_j = B_j$. This preprocessing step significantly decreases the number of calls of $\mathrm{PrU}(\cdot)$ in recursion, and after introducing a specific ordering of the input events $B_1, ..., B_N$, the efficiency of the sum-of-disjoint-products can be further enhanced [10]. Despite these heuristic improvements, Eq. (2.6) remains intractable for large N_c, say more than a hundred. Also note that Eqs. (2.4)-(2.7) are applicable for computing the union of γ-MPs in Eq. (2.3), by simply replacing the events $\{A_i\}_{i=1}^{N_c}$ in the equations with $\{A_i^*\}_{i=1}^{N_r}$.

For connectivity-based problems, more efficient algorithms exist for computing network reliability. Some are based on the domination theory by observing that many terms in Eq. (2.4) are equivalent

and hence can be canceled out in advance [126, 125, 2]; others rely on the Boolean algebra, whereby the number of terms in Eq. (2.6) can be significantly reduced [56, 1, 99, 136]. Additionally, given all minimal cuts, Provan and Ball [115] compute two-terminal reliability in time polynomial in the number of minimal cuts; given all minimal paths, Ball and Nemhauser [13] calculate all-terminal reliability in time polynomial in the number of minimal paths. However, all the above-mentioned methods run in time exponential in the number of network components, which is as expected since the connectivity-based reliability assessment is proven to be NP-hard [11].

Note that, although our primary focus in this section is on independent inputs, the cut(path)-based method can be extended to address dependent component failures [105].

2.2.3 State space decomposition (SSD) methods

2.2.3.1 State space decomposition (SSD)

The SSD method was originally introduced by Doulliez and Jamoulle [48] for computing the probability of fulfilling demands for multiple terminal nodes within a stochastic network. Unfortunately, the decomposition strategy described in that work is not always accurate, and Alexopoulos [4] addressed and corrected this issue in the context of two-terminal maximum flow problems. The modified algorithm was subsequently refined by Jane and Laih [76].

Although the SSD method can also be extended to a broad spectrum of other network problems [73, 42], we only focus on the two-terminal maximum flow problem for the sake of illustration. Recall that the goal of maximum flow problems is to compute the probability that a specified γ unit of flow cannot be delivered from the source node to the sink node in a stochastic graph $\mathcal{G}(\mathcal{E}, \mathcal{V})$. $\mathcal{E}$ and $\mathcal{V}$ denote the edge set and node set, respectively. The capacity of each d-th edge is modeled as a categorical distributed variable, taking n_d states $0 = s_{d,1} < s_{d,2} < \cdots < s_{d,n_d}$ with probability $p_{d,1}, \cdots, p_{d,n_d}$, respectively. Also, we assume the states of different edges are independent. Let $\boldsymbol{X}$ collect the state of all n edges with state space $\Omega_{\boldsymbol{X}} \triangleq \{\boldsymbol{x} | x_d \in \boldsymbol{s}_d \triangleq \{s_{d,1}, \cdots, s_{d,n_d}\}, d = 1, \cdots, n\}$ and probability mass function $p_{\boldsymbol{X}}(\boldsymbol{x})$. The function $g^{(\mathrm{mf})}(\boldsymbol{x})$ computes the maximum flow from the source node to the sink node given $\boldsymbol{x}$, a realization of the random capacity $\boldsymbol{X}$. We term a domain $\mathcal{I}$ infeasible if it consists of only infeasible states, for which $g^{(\mathrm{mf})}(\boldsymbol{x}) < \gamma$ holds, whereas a feasible domain $\mathcal{F}$ comprises only feasible states with $g^{(\mathrm{mf})}(\boldsymbol{x}) \geq \gamma$. An unspecified domain $\mathcal{U}$ does not fit into either of the two categories and contains unclassified states.

The SSD method progressively refines the lower and upper bound of $\Pr(g^{(\mathrm{mf})}(\boldsymbol{x}) < \gamma)$ through iteratively removing feasible and infeasible domains from the current unspecified state space. Specifically, the method initiates with the state space $\Omega_{\boldsymbol{X}}$, i.e., $\mathcal{U}^{(1)} = \Omega_{\boldsymbol{X}}$. The partition of the unspecified domain $\mathcal{U}^{(l)}$ at the l-th level hinges on identifying critical edge capacities $\boldsymbol{x}^o \in \mathcal{U}^{(l)}$ and $\boldsymbol{x}^* \in \mathcal{U}^{(l)}$ that ensure: (1) $g^{(\mathrm{mf})}(\boldsymbol{x}) \geq \gamma$ if $\boldsymbol{x} \geq \boldsymbol{x}^o$, i.e., if $x_d \geq x_d^o$ holds for each $d \in \{1, \cdots, n\}$. $\boldsymbol{x}^o$ is, in fact, a γ-MP described in Subsection 2.2.2, but here we adopt the notation used in the original paper [48];

(2) $g^{(\mathrm{mf})}(\boldsymbol{x}) < \gamma$ if $x_d < x_d^*$ for some d. Subsequently, the following sets are defined:

$$\begin{aligned} \mathcal{F} &\triangleq \{\boldsymbol{x} \in \mathcal{U}^{(l)} | \boldsymbol{x} \geq \boldsymbol{x}^o\}, \\ \mathcal{I} &\triangleq \bigcup_{d=1}^{n} \{\boldsymbol{x} \in \mathcal{U}^{(l)} | x_d < x_d^*\}, \\ \mathcal{U} = \mathcal{U}^{(l)} \cap \overline{\mathcal{I}} \cap \overline{\mathcal{F}} &\triangleq \bigcup_{d=1}^{n} \{\boldsymbol{x} \in \mathcal{U}^{(l)} | \boldsymbol{x} \geq \boldsymbol{x}^*, x_d < x_d^o\}. \end{aligned} \tag{2.8}$$

The upper bound of $p_f \triangleq \Pr(g^{(\mathrm{mf})}(\boldsymbol{x}) < \gamma)$ is then refined by substracting $\Pr(\mathcal{F})$, and the lower bound is increased by $\Pr(\mathcal{I})$. The same procedure is applied to decompose $\mathcal{U}^{(l+1)} = \mathcal{U}$.

In practice, the SSD method operates on hyperrectangular domains. A hyperrectangle in $\Omega_{\boldsymbol{X}}$, denoted as box $\left[\boldsymbol{\alpha} \triangleq (\alpha_1, \cdots, \alpha_n), \boldsymbol{\beta} \triangleq (\beta_1, \cdots, \beta_n)\right]$, is defined as

$$\mathrm{box}\left[\boldsymbol{\alpha}, \boldsymbol{\beta}\right] = \{\boldsymbol{x} \in \Omega_{\boldsymbol{X}} | \alpha_d \leq x_d \leq \beta_d, \quad d = 1, \cdots, n\}, \tag{2.9}$$

where α_d and β_d are constrained to values within the set $\boldsymbol{s}_d$. The probability of box $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ is computed by

$$\Pr(\mathrm{box}\left[\boldsymbol{\alpha}, \boldsymbol{\beta}\right]) = \prod_{d=1}^{n} \sum_{j=1}^{n_d} \mathbb{I}\{\alpha_d \leq s_{d,j} \leq \beta_d\} p_{d,j}, \tag{2.10}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function.

Partitioning the unspecified domain $\mathcal{U}^{(l)}$ is significantly simplified when $\mathcal{U}^{(l)}$ is a hyperrectangle, i.e., $\mathcal{U}^{(l)} = \mathrm{box}\left[\boldsymbol{\alpha}, \boldsymbol{\beta}\right]$. In this context, the critical edge capacity, $\boldsymbol{x}^o$, can be determined by the following three steps [48]: (1) Create a fictitious node and connect it to the sink node. (2) Calculate the maximum flow from the source node to the sink node, with capacities $\boldsymbol{\beta}$ for the edges in $\mathcal{E}$ and capacity γ for the fictitious edge that links the fictitious node to the sink node. (3) The flow in each edge is then assigned as $\boldsymbol{x}^o$. Selecting $\boldsymbol{x}^*$, however, is more complex. The approach proposed in Doulliez and Jamoulle [48] was found to be inaccurate and subsequently improved by Alexopoulos [4]. Alternatively, $\boldsymbol{x}^*$ can be simply selected as $\boldsymbol{\alpha}$, which is commonly referred to as feasible-based partitioning in the literature [42, 112], as it leads to an empty infeasible domain $\mathcal{I}$ at each iteration. Now, the feasible, infeasible, and unspecified domains defined in Eq. (2.8) can be written as

$$\begin{aligned} \mathcal{F} &= \mathrm{box}\left[\boldsymbol{x}^o, \boldsymbol{\beta}\right], \\ \mathcal{I} &= \bigcup_{d=1}^{n} \mathrm{box}\left[\boldsymbol{\alpha}, (\beta_1, \cdots, \beta_{d-1}, \underline{x_d^*}, \beta_{d+1}, \cdots, \beta_n)\right], \\ \mathcal{U} &= \bigcup_{d=1}^{n} \mathrm{box}\left[\boldsymbol{x}^*, (\beta_1, \cdots, \beta_{d-1}, \underline{x_d^o}, \beta_{d+1}, \cdots, \beta_n)\right], \end{aligned} \tag{2.11}$$

where $\underline{x_d^*}$ (resp. $\underline{x_d^o}$) is the largest state in $\boldsymbol{s}_d$ that is smaller than x_d^* (resp. x_d^o). However, we cannot proceed to partition $\mathcal{U}$ using the above strategy unless it is hyperrectangular. Hence, further decomposition of $\mathcal{U}$ into disjoint hyperrectangles is necessary. Let

$$\mathcal{U}_j \triangleq \{\boldsymbol{x} \in \mathcal{U}^{(l)} | x_j^* \leq x_j < x_j^o; x_d \geq x_d^o \text{ if } d < j; x_d \geq x_d^* \text{ if } d > j\}. \tag{2.12}$$

It holds that $\mathcal{U} = \bigcup_{j=1}^{n} \mathcal{U}_j$ and $\mathcal{U}_i \cap \mathcal{U}_j = \emptyset$ for $i \neq j$. In other words, $\mathcal{U}_j, j = 1, \cdots, n$ form a partition of $\mathcal{U}$. More importantly, $\mathcal{U}_j$ is hyperrectangular, and can be written as

$$\mathcal{U}_j = \text{Box}\left[(x_1^o, \cdots, x_{j-1}^o, x_j^*, x_{j+1}^*, \cdots, x_n^*)(\beta_1, \cdots, \beta_{j-1}, \underline{x_j^o}, \beta_{j+1}, \cdots, \beta_n)\right]. \tag{2.13}$$

Similar decomposition can be applied to get the disjoint partition of $\mathcal{I}$, denoted as $\mathcal{I}_j, j = 1, \cdots, n$, and it holds that $\Pr(\mathcal{I}) = \sum_{j=1}^{n} \Pr(\mathcal{I}_j)$. $\Pr(\mathcal{I}_j)$ and $\Pr(\mathcal{F})$ can be calculated through Eq. (2.10). These probabilities are then utilized to refine both the lower and upper bounds of the failure probability p_f. The hyperrectangles $\mathcal{U}_j, j = 1, \cdots, n$ are added to the set of unspecified domains. From this set, $\mathcal{U}^{(l+1)}$ is selected for decomposition in the next level.

The issue is that for each unspecified hyperrectangle, n new ones are generated in the next iteration. Although some of them will be specified (i.e., empty, feasible, or infeasible) or duplicated, the total number of unspecified hyperrectangles still grows exponentially. When organizing these hyperrectangles in a tree, devising an effective partition strategy (as it directly impacts the tree's structure) and selecting an appropriate order for traversing the tree become essential for enhancing the efficiency of the SSD method [112]. For large systems with moderate component failure probabilities, it often requires a large amount of time to narrow the bounds (especially the upper bound) to a meaningful value. To mitigate this issue, one can terminate the SSD method prematurely and employ a sampling-based method to explore the remaining unspecified domain [73].

2.2.3.2 Recursive decomposition methods

In the field of connectivity-based reliability assessment, the idea of iteratively partitioning the state space was independently proposed by Dotson and Gbien [47]. The algorithm was subsequently optimized by encoding the graph with its adjacency matrix [139]. The performance of both Dotson's and modified Dotson's algorithms can be enhanced with parallel computing [44]. Building upon these prior contributions, Li and He [91] proposed the recursive decomposition method for analyzing K-terminal connectivity of stochastic networks. This method is applicable to both direct and indirect graphs and accounts for node failure. However, similar to the SSD method, the reliability bounds may converge slowly in large networks. To mitigate this issue, Lim and Song [93] prioritized the subgraph that has the largest decomposition factor and decomposed it with its most reliable path (or cut). Recent developments of the recursive decomposition method include extensions to address dependent component failure [69] and integrations with network clustering techniques [89].

In the following, we introduce the recursive decomposition method in the context of two-terminal connectivity problems considering only independent edge failure. We denote the state of all n edges as $\boldsymbol{x} = \{x_1, \cdots, x_n\}$, where each d-th edge can either be failed or operational with probability p_d and $1 - p_d$, respectively. The algorithm initializes with the original graph, and its probability is set to one, i.e., $\mathcal{G}^{(1)} = \mathcal{G}(\mathcal{V}, \mathcal{E})$ and $\Pr(\mathcal{G}^{(1)}) = 1$. At the l-th iteration, the recursive decomposition method first identifies a minimal path within the current graph $\mathcal{G}^{(l)}$ through the breath-first-search algorithm (or Dijkstra's algorithm [93]). Let $\boldsymbol{a} \triangleq (a_1, \cdots, a_k)$ collect the edge indices of the selected

minimal path. The following subgraphs are then generated:

$$\begin{aligned}
\mathcal{G}_0^{(l)} &\triangleq \mathcal{G}^{(l)} + (x_{a_1}, \cdots, x_{a_k}) \\
\mathcal{G}_1^{(l)} &\triangleq \mathcal{G}^{(l)} + (\overline{x_{a_1}}) \\
\mathcal{G}_2^{(l)} &\triangleq \mathcal{G}^{(l)} + (x_{a_1}, \overline{x_{a_2}}) \\
&\vdots \\
\mathcal{G}_k^{(l)} &\triangleq \mathcal{G}^{(l)} + (x_{a_1}, \cdots, x_{a_{k-1}}, \overline{x_{a_k}}).
\end{aligned} \tag{2.14}$$

$\mathcal{G}^* + \boldsymbol{b}$ denotes a graph induced by $\mathcal{G}^*$ and $\boldsymbol{b}$, where $\boldsymbol{b}$ is a boolean vector that specifies the states of certain edges in $\mathcal{G}^*$. For instance, the graph induced by $\mathcal{G}^*$ and $\boldsymbol{b} = (x_1, \overline{x_4})$ is obtained by contracting the first edge and then removing the fourth one from $\mathcal{G}^*$. Since each edge fails independently, the probability of the graph $\mathcal{G}^* + \boldsymbol{b}$ can be calculated as $\Pr(\mathcal{G}^*) \cdot \prod_{d=1}^{n} (1 - p_d)^{\mathbb{I}\{x_d \in \boldsymbol{b}\}} \cdot (p_d)^{\mathbb{I}\{\overline{x_d} \in \boldsymbol{b}\}}$. If any of $\mathcal{G}_0^{(l)}, \cdots, \mathcal{G}_k^{(l)}$, is already connected or disconnected, it is added to the path set or cut set, respectively. The remaining graphs are added to the set of undecomposed graphs, from which $\mathcal{G}$ is selected for decomposition in the next iteration.

The recursive decomposition method outputs the lower and upper bounds of the reliability at each iteration. Specifically, the lower bound is determined by adding the probabilities of the graphs in the path set, while the upper bound is calculated by subtracting the summation of the probabilities of cut set graphs from one. The convergence rate of the bounds and the space complexity of the algorithm depend on both the decomposition order and rules. This includes the selection of $\mathcal{G}^{(l+1)}$, the identification of the minimal path $\boldsymbol{a}$, and the order of arranging elements within $\boldsymbol{a}$. A set of heuristic strategies have been suggested in the literature [93, 89, 112].

Each subgraph generated by the recursive decomposition method can be interpreted as a hyper-rectangle domain, which is illustrated in Fig. 2.3. Specifically, connected subgraphs in the path set are disjoint feasible domains; disconnected ones in the cut set represent disjoint infeasible domains; unclassified graphs are unspecified domains. In this way, the recursive decomposition method can be interpreted as a feasible-based SSD method.

Each subgraph in the recursive decomposition method can also be encoded with a Boolean function, as shown in Fig. 2.3. From this viewpoint, the recursive decomposition method is an extension of the BDD. Both methods rely on iteratively decomposing the Boolean structure function. However, in BDD, the decomposition is rooted in Shannon's decomposition formula, whereas in the recursive decomposition method, it is related to a minimal path. In particular, Eq. (2.14) can be expressed as

$$f = x_{a_1} x_{a_2} \cdots x_{a_k} \cdot 1 + \overline{x_{a_1}} \cdot f|_{x_{a_1}=0} + \cdots + x_{a_1} x_{a_2} \cdots x_{a_{k-1}} \overline{x_{a_k}} \cdot f|_{x_{a_1}=1, \cdots, x_{a_{k-1}}=1, x_{a_k}=0}, \tag{2.15}$$

where $f|x_{a_1} = 0$ (resp. $f|x_{a_1} = 1$) denote the Boolean function f with x_{a_1} being false (resp. true).

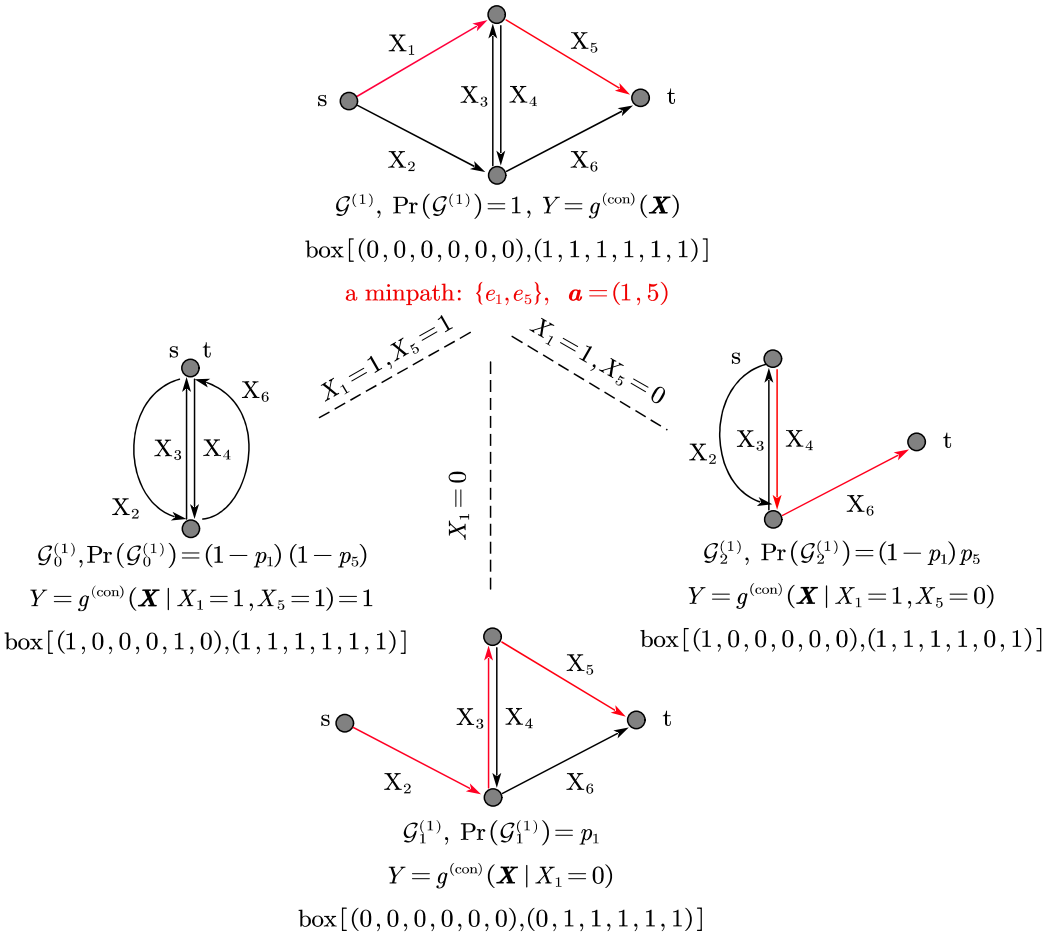


Figure 2.3: The initial iteration of the recursive decomposition method. One first specifies a minimal path, e.g., $\{e_1, e_5\}$, and decomposes the original graph $\mathcal{G}^{(1)}$ according to it, resulting in three subgraphs: $\mathcal{G}_0^{(1)}$, $\mathcal{G}_1^{(1)}$, and $\mathcal{G}_2^{(1)}$. Each subgraph corresponds to a hyperrectangle in the SSD method and can also be interpreted as a Boolean function. For instance, the subgraph $\mathcal{G}_0^{(1)}$ represents the hyperrectangle between $(1,0,0,0,1,0)$ and $(1,1,1,1,1,1)$, and represents the Boolean function $g^{(\mathrm{con})}(\boldsymbol{X} \mid X_1 = 1, X_5 = 1)$, which is always true since $\mathcal{G}_0^{(1)}$ is already connected. In contrast, $\mathcal{G}_1^{(1)}$ and $\mathcal{G}_2^{(1)}$ remain undetermined and will undergo decomposition in subsequent iterations. The lower and upper bound of the failure probability after the first iteration are, therefore, equal to 0 and $1 - \Pr(\mathcal{G}_0^{(1)})$, respectively.

2.3 Sampling techniques

Compared to the non-sampling methods discussed in the previous section, sampling-based methods are often more general, even applicable to black-box models, albeit yielding less reliable estimates. In the following section, we introduce popular sampling techniques for network reliability assessment. These include crude Monte Carlo, counting-based methods, creation-process-based approaches, along with two common variance reduction techniques: splitting and importance sampling. Again, we introduce these methods separately and leave the detailed comparison to Section 2.5.

2.3.1 Crude Monte Carlo

Crude MCS is the most widely employed sampling-based approach for estimating expectations, especially in high dimensions. Let h be a deterministic function with input $\boldsymbol{X}$, where $\boldsymbol{X} = (X_1, \cdots, X_n)$ is a n-dimensional random vector whose distribution, referred to as $p_{\boldsymbol{X}}(\boldsymbol{x})$, is supported over $\Omega_{\boldsymbol{X}}$. Crude MCS then estimates the expectation $\mathbb{E}_{p_{\boldsymbol{X}}}(h(\boldsymbol{X}))$ by its sample mean. That is:

$$\widehat{p}^{(MCS)} \triangleq \frac{1}{N}\sum_{k=1}^{N} h(\boldsymbol{x}^{(k)}), \quad \boldsymbol{x}^{(k)} \sim p_{\boldsymbol{X}}(\boldsymbol{x}), k = 1, \cdots, N. \tag{2.16}$$

Hereafter, we adhere to the convention that $\widehat{p}$ denotes an estimator. In network reliability problems, the input vector $\boldsymbol{X}$ commonly represents the states of network components, and $h(\boldsymbol{x})$ often serves as an indicator function that equals 1 when the network performance $g(\boldsymbol{x})$ exceeds or falls below the failure threshold γ, i.e., the network is failed, and 0 otherwise. In this context, the expectation $\mathbb{E}_{p_{\boldsymbol{X}}}(h(\boldsymbol{X}))$ is exactly the failure probability p_f:

$$p_f = \mathbb{E}_{p_{\boldsymbol{X}}}\left(\mathbb{I}\{\boldsymbol{X} \in F\}\right). \tag{2.17}$$

Recall that F denotes the failure domain. In the context of connectivity-based problems, the domain is defined as $\{\boldsymbol{x} \mid g^{(\text{conn})} = 0\}$; for maximum-flow problems, it corresponds to $\{\boldsymbol{x} \mid g^{(\text{mf})} < \gamma\}$; in power flow problems, it is characterized by $\{\boldsymbol{x} \mid g^{(\text{bz})} \geq \gamma\}$.

The MCS estimator $\widehat{p}^{(MCS)}$ is unbiased with the following variance:

$$\text{Var}(\widehat{p}^{(MCS)}) = \frac{\text{Var}(h(\boldsymbol{X}))}{N} = \frac{\text{Var}(\mathbb{I}\{\boldsymbol{X} \in F\})}{N} = \frac{p_f(1-p_f)}{N}. \tag{2.18}$$

The coefficient of variation of the MCS estimator is proportional to $\sqrt{\frac{1-p_f}{p_f}}$. In addition, note that $\sum_{k=1}^{N} h(\boldsymbol{x}^{(k)}) = \sum_{k=1}^{N} \mathbb{I}\{\boldsymbol{x}^{(k)} \in F\}$ is the sum of a sequence of N independent Bernoulli random variables, each with a probability p_f of being one. By applying the Chernoff bound [36], one can prove that:

$$\Pr\left(\frac{|\widehat{p}^{(MCS)} - p_f|}{p_f} \geq \epsilon\right) \leq 2\exp\left(-\frac{N\epsilon^2 p_f}{3}\right), \tag{2.19}$$

where ϵ is a user-specified tolerance. As a corollary, generating $N \geq \frac{-3\ln\frac{\delta}{2}}{\epsilon^2 p_f} \triangleq N^*$ samples is sufficient to bound the absolute relative error within ϵ with probability at least $1-\delta$, which is derived by

ensuring $1 - \Pr\left(\frac{|\widehat{p}^{(MCS)} - p_f|}{p_f} \geq \epsilon\right) \geq 1 - 2\exp\left(-\frac{N\epsilon^2 p_f}{3}\right) \geq 1 - \delta$. This is also known as the (ϵ, δ)-approximation in statistics [113]. Unfortunately, the sample size N^* is unknown in practice due to its dependence on the unknown quantity of interest p_f, but if a lower bound of p_f, denoted as B, is given, the sample size N can be determined conservatively as $\frac{-3\ln\frac{\delta}{2}}{\epsilon^2 B}$. In addition, there exist various modifications of crude MCS that offer assured (ϵ, δ)-approximations [113]. The time taken by these algorithms to reach the stopping criterion is random, and they are categorized as Las Vegas algorithms in computer science.

Eqs. (2.18) and (2.19) indicate that to attain a bounded coefficient of variation or an (ϵ, δ)-approximation, the necessary number of samples in MCS is of the order $O\left(\frac{1}{p_f}\right)$. When evaluating the network performance g is often computationally expensive; crude MCS is therefore infeasible for rare event estimation. In such cases, different variance reduction techniques need to be incorporated.

Multi-level Splitting and adaptive importance sampling are two primary variance reduction techniques intensively investigated in structural reliability analysis. These methods aim to guide the sampling procedure toward the failure domain by either replicating and advancing elite samples, i.e., samples located closer to the failure domain, or by sampling from a series of intermediate distributions that progressively converge toward the optimal importance sampling distribution. They both work in an adaptive manner, relying on a sequence of levels that may either be predefined or adaptively selected during the simulation and involve a negligible computational overhead when dealing with complex models. Consequently, their computational cost can be accurately approximated by the number of calls to the performance function g. Based on these similarities, we categorize multi-level splitting and importance sampling as adaptMCS methods in the remaining part of this thesis. Note that in network reliability assessment, where the network performance is often discrete, the adaptMCS methods initially proposed for continuous performance functions may encounter difficulties, resulting in substantial errors. To circumvent this issue, a potential approach is to introduce the creation process that substitutes the discrete performance $g(\boldsymbol{X})$ with a continuous critical time T^*. Subsequently, the adaptMCS methods are applied to the transformed performance functions. However, the transformed performance function is more difficult and expensive to evaluate, often requiring multiple calls to the original function g in order to obtain the critical order $[\pi]$. This further motivates the idea explored in this thesis, which is the extension of adaptMCS methods to network reliability assessment without incorporating the creation process. A detailed introduction to the creation process, multi-level splitting, and adaptive importance sampling will be given later in this section.

In addition to the adaptive approaches, alternative variance-reduction techniques are available for estimating rare events in networks. These include dagger sampling [84], sequence construction [50], MCS with restricted sample space [83, 54, 55, 134, 107, 5], the coverage method [81], stratified sampling [70]. A comprehensive overview of these methods can be found in Brown et al. [20].

2.3.2 Counting-based method

The counting-based method [49, 113] is proposed for solving the connectivity-based problem, in which we calculate the probability that a group of K nodes is disconnected in a random graph $\mathcal{G}(\mathcal{V} \triangleq \{v_1, \cdots, v_m\}, \mathcal{E} \triangleq \{e_1, \cdots, e_n\})$, with $\mathcal{V}$, $\mathcal{E}$ being the node set and edge, respectively. There

are in total n edges and m nodes, and the indices of K terminal nodes are collected in the set $\mathcal{K}$.

All nodes are assumed to be perfect, but each i-th edge can fail independently with probability p_i. Hence, the state of all n edges can be collected in a binary vector $\boldsymbol{X}$, which is independently Bernoulli distributed. The connectivity of the graph is indicated by $g^{(\text{conn})}(\boldsymbol{x}) : \boldsymbol{x} \in \{0,1\}^n \to \{0,1\}$, which is also known as the structure function or performance function. Specifically, $g^{(\text{conn})}(\boldsymbol{x}) = 0$ implies that $\boldsymbol{x}$ is a failure state that disconnects the nodes indexed in $\mathcal{K}$.

To introduce the basic idea behind this counting-based approach, we first consider $p_i = 0.5$ for each $1 \leq i \leq n$. In this context, the system failure probability can be expressed as the ratio of the number of states that disconnect the network, denoted as $|F|$, to the number of total states, denoted as $|\Omega_{\boldsymbol{X}}|$. Here, the cardinality of a set is denoted as $|\cdot|$. That is

$$p_f = \frac{|F|}{|\Omega_{\boldsymbol{X}}|}. \tag{2.20}$$

$|\Omega_{\boldsymbol{X}}|$ is known to be 2^n for a n-component binary system, so the failure probability calculation degenerates into counting the failure states. After encoding the complement of the structure function, i.e., $\neg g^{(\text{conn})}(\boldsymbol{x}) = 1 - g^{(\text{conn})}(\boldsymbol{x})$, as a Σ_1^1 formula, various projected counting techniques, which have been intensively investigated in Boolean satisfiability analysis, can be leveraged to produce a probably approximately correct (PAC) estimate of $|F|$. Consequently, the resulting failure probability estimator is also PAC.

The concepts of conjunctive normal form (CNF) and Σ_1^1 formula are essential for understanding the counting-based approach and hence are introduced in the following. In Boolean logic, there are three basic operations: logic *and* $\wedge$ (also known as conjunction), *or* $\vee$ (disjunction), and *not* $\neg$ (negation). A Boolean function $f(\boldsymbol{x})$ is in CNF if it is written as a conjunction of disjunctions of Boolean variables or their negations; otherwise put, in CNF, disjunctions must be nested in a conjunction, and negations must be nested in a disjunction or conjunction. For example, $\neg(x_1 \vee x_2)$ is not in CNF since $\neg$ is outside $\vee$, but it can be rewritten as $\neg x_1 \wedge \neg x_2$, which is in CNF. A Boolean function $f(\boldsymbol{x})$ is in Σ_1^1 form if it can be written as $\vee_{\boldsymbol{z} \in \{0,1\}^m}[\varphi(\boldsymbol{x}, \boldsymbol{z})]$, where $\boldsymbol{z} \triangleq \{z_1, ..., z_m\}$ is a set of m auxiliary Boolean variables, and φ is a CNF expression over $\boldsymbol{x}$ and $\boldsymbol{z}$. In other words, $f(\boldsymbol{x}) = 0$ if and only if $\varphi(\boldsymbol{x}, \boldsymbol{z}) = 0$ holds for every $\boldsymbol{z} \in \{0,1\}^m$.

Paredes et al. [113] introduce a Σ_1^1 formula for counting K-terminal reliability, termed the $\mathcal{K}-$RelNet. Let $\boldsymbol{z} \triangleq \{z_1, ..., z_m\}$ denote m auxiliary variables, each associated with a node in $\mathcal{V}$, and $(i_0, i_1) \in \{1, \cdots, m\}$ be the index of the end nodes of the i-th edge. The $\mathcal{K}-$RelNet formulation reads

$$C_i = (\neg z_{i_0} \vee \neg x_i \vee z_{i_1}) \wedge (z_{i_0} \vee \neg x_i \vee \neg z_{i_1}) = \begin{cases} 1, & z_{i_0} = z_{i_1} \\ \neg x_i, & z_{i_0} \neq z_{i_1} \end{cases} \tag{2.21}$$

$$\varphi(\boldsymbol{x}, \boldsymbol{z}) = \underbrace{(\vee_{j \in \mathcal{K}} z_j) \wedge (\vee_{k \in \mathcal{K}} \neg z_k)}_{\textbf{1-st term}} \wedge \underbrace{(\wedge_{i=1}^{n} C_i)}_{\textbf{2-nd term}}. \tag{2.22}$$

In the following, we add a new interpretation of this formulation and explain why it works. Let us first assign each node in the graph a color according to the auxiliary vector $\boldsymbol{z}$. For instance, we can color the j-th node black when $z_j = 1$ and white otherwise. Based on Eq. (2.21), C_i is true in the following two cases: (1) The i-th edge is failed. (2) The i-th edge is functional, and the color of its two end nodes is the same. Hence, to let the second term in Eq. (2.22) be true, the nodes connected

must share the same color. However, if there is only one color for $\mathcal{K}$, the first term in Eq. (2.22) will become zero. This means that if the nodes indexed in $\mathcal{K}$ are all connected, i.e., $g^{(\text{conn})}(\boldsymbol{x}) = 1$, there is no way to make the auxiliary function $\varphi(\boldsymbol{x}, \boldsymbol{z})$ true. The complement of the structure function, i.e., $\neg g^{(\text{conn})}(\boldsymbol{x})$, can therefore be written in Σ_1^1 form with Eq. (2.22). Subsequently, the PAC estimate for $|F|$ is derived using projected counting techniques.

The counting-based approach is extended to encompass arbitrary but still independent p_i, where $i = 1, \cdots, n$ through a weighted-to-unweighted transformation. In this context, an unweighted edge implies the edge's failure probability is equal to 0.5; otherwise, the edge is weighted. The basic idea of the transformation is to substitute each weighted edge with an unweighted series-parallel system whose failure probability is approximately equal to that of the original edge. This transformation hinges on representing p_i as a binary vector with κ-bits, and the efficiency and accuracy of the following counting process are sensitive to the choice of κ. Selecting a small value for κ can lead to a non-negligible truncate error, whereas opting for a large value will significantly increase the computation cost. This issue becomes critical when p_i is small. Besides, encoding arbitrary dependent p_i is challenging in the counting-based approach.

2.3.3 Creation process

For rare event estimation, crude MCS can be infeasible, and variance reduction techniques are required. In the field of network reliability assessment, many of the variance reduction techniques are built on a graph evolution process termed the creation process [53, 52], in which the network component is assumed to be repaired sequentially from a completely damaged network with no functional component. After introducing a random repair time for each component with a proper distribution, the static network reliability problem can be transformed into a dynamic one. Instead of estimating the network reliability, it suffices to compute the probability that the network becomes functional before a unit of time. As a result, the often discrete network performance is smoothed by a continuous critical time, which indicates the moment the network first becomes functional. Various variance reduction techniques that were initially proposed for continuous structural performance can be leveraged for computing the network reliability, e.g., the multi-level splitting [17, 18, 104, 24] and CE method [71]. Moreover, the creation process provides a natural way of defining permutation Monte Carlo simulation (pMCS) [50, 53], which independently generates the permutation of network components and calculates the reliability conditional on the generated permutation. The accuracy of the pMCS can be further enhanced through a merging process [53]. Note that it is also possible to design a destruction process where components are removed from an intact network sequentially, and the objective is to estimate the probability the network fails after a unit of time.

2.3.3.1 Creation process for systems with binary inputs

2.3.3.1.1 The initial idea

The creation process was first introduced by Elperin et al. [53] for solving connectivity-based problems, where the backbone model is an undirected graph (or network) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \cdots, v_m\}$ and $\mathcal{E} = \{e_1, \cdots, e_n\}$ denoting the node set and edge set, respectively. The nodes are

assumed to be perfect, but each i-th edge e_i, where $i = 1, \cdots, n$, can either fail or not fail with probability p_i and $1 - p_i$, respectively. The state of each edge, denoted as X_i for $i = 1, \cdots, n$, is modeled as an independent Bernoulli random variable, $\mathrm{Ber}(x_i; 1-p_i)$, with 1 denoting the functional state and 0 otherwise. The quantity of interest is the probability that a set of K nodes in the graph is disconnected, denoted as p_f. The index set of K terminal nodes is denoted as $\mathcal{K}$.

This static model can be reformulated as a graph evolution process referred to as the creation process, wherein the edge-set and, hence, also the graph (or network) evolve over an artificial time t. Let $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ denote this process, where $\mathcal{E}(t)$ collects the functional edges up to time t. Initially, all edges are failed, so $\mathcal{E}(t = 0) = \emptyset$. Each edge then undergoes an independent repair at a random time T_i, characterized by a continuous cumulative distribution function (CDF) $F_{T_i}(t) \triangleq \Pr(T_i \leq t), t \geq 0$ such that $F_{T_i}(t = 1) = 1 - p_i$. Once the edge is repaired, it is permanently added into $\mathcal{E}(t)$. In this way, the snapshot of the graph at $t = 1$, i.e., $\mathcal{G}(t = 1)$, aligns with the original static model since each edge in $\mathcal{G}(t = 1)$ is down independently with probability p_i. Moreover, since the repair of additional edges will not disconnect the network, i.e., connectivity is a coherent network performance, it suffices to compute the probability that the critical time T^*, when the graph $\mathcal{G}(t)$ first becomes connected, is larger than 1; otherwise put, $p_f = \Pr(T^* > 1)$.

Depending on the choice of $F_{T_i}(t)$, different creation processes can be defined [17]. In most cases, however, an exponential distribution is favored due to its memoryless property, which significantly simplifies the computation complexity. In the remainder of this section, we will focus on this particular case. Specifically, the repair time of the i-th edge, T_i, takes the following CDF:

$$\begin{aligned} \lambda_i &= -\ln(p_i), \\ F_{T_i}(t) &= 1 - \exp(-\lambda_i \cdot t), t \geq 0. \end{aligned} \tag{2.23}$$

Also, T_i and T_j are independent for $i \neq j$. It is evident that the probability T_i exceeding 1 is given by $\Pr(T_i > 1) = 1 - F_{T_i}(1) = p_i$.

The repair time of all edges, $t_1, \cdots, t_n$, specifies the order, $\boldsymbol{\pi} \triangleq (\pi_1, \cdots, \pi_n)$, in which the edges are repaired. Specifically, the π_1-th edge, e_{π_1}, is repaired first, followed by e_{π_2}, e_{π_3}, and so forth. The repair time also determines a realization of the creation process $\widetilde{\mathcal{G}}(t)$, which remains in $\emptyset$ until t_{π_1} and in $\mathcal{G}(\mathcal{V}, \{e_{\pi_1}, \cdots, e_{\pi_l}\})$ between t_{π_l} and $t_{\pi_{l+1}}$ for $l = 1, \cdots, n$, with $t_{\pi_{n+1}} = \inf$. In addition, there must be a critical order, denoted as $[\boldsymbol{\pi}]$, such that $\mathcal{G}(\mathcal{V}, \{e_{\pi_1}, \cdots, e_{\pi_{[\boldsymbol{\pi}]}}\})$ is connected and $\mathcal{G}(\mathcal{V}, \{e_{\pi_1}, \cdots, e_{\pi_{[\boldsymbol{\pi}]-1}}\})$ is not. In other words, the repair of $e_{\pi_{[\boldsymbol{\pi}]}}$ transforms the graph from a disconnected state to a connected state. Therefore, by definition, the critical time t^* equals the time the $\pi_{[\boldsymbol{\pi}]}$-th edge is repaired, that is, $t_{\pi_{[\boldsymbol{\pi}]}}$. Let $\Delta t_l \triangleq t_{\pi_{l+1}} - t_{\pi_l}$ denote the sojourn time of $\widetilde{\mathcal{G}}(t)$ in $\mathcal{G}(\mathcal{V}, \{e_{\pi_1}, \cdots, e_{\pi_l}\})$ and $\Delta t_0 = t_{\pi_1}$ be the sojourn time of $\widetilde{\mathcal{G}}(t)$ in $\emptyset$. The critical time t^* can be rewritten as $\sum_{l=0}^{[\boldsymbol{\pi}]-1} \Delta t_l$. Note that since $t_1, \cdots, t_n$ are not fixed but generated from the independent exponential distribution described by Eq. (2.23), $\boldsymbol{\pi}, [\boldsymbol{\pi}], \Delta t_l, t^*$ are also random. These random variables are denoted as $\boldsymbol{\Pi}, [\boldsymbol{\Pi}], \Delta T_l, T^*$ in the remaining part of this section. A realization of the creation process for the toy example in Subsection 2.2.2.1 is illustrated in Fig. 2.4.

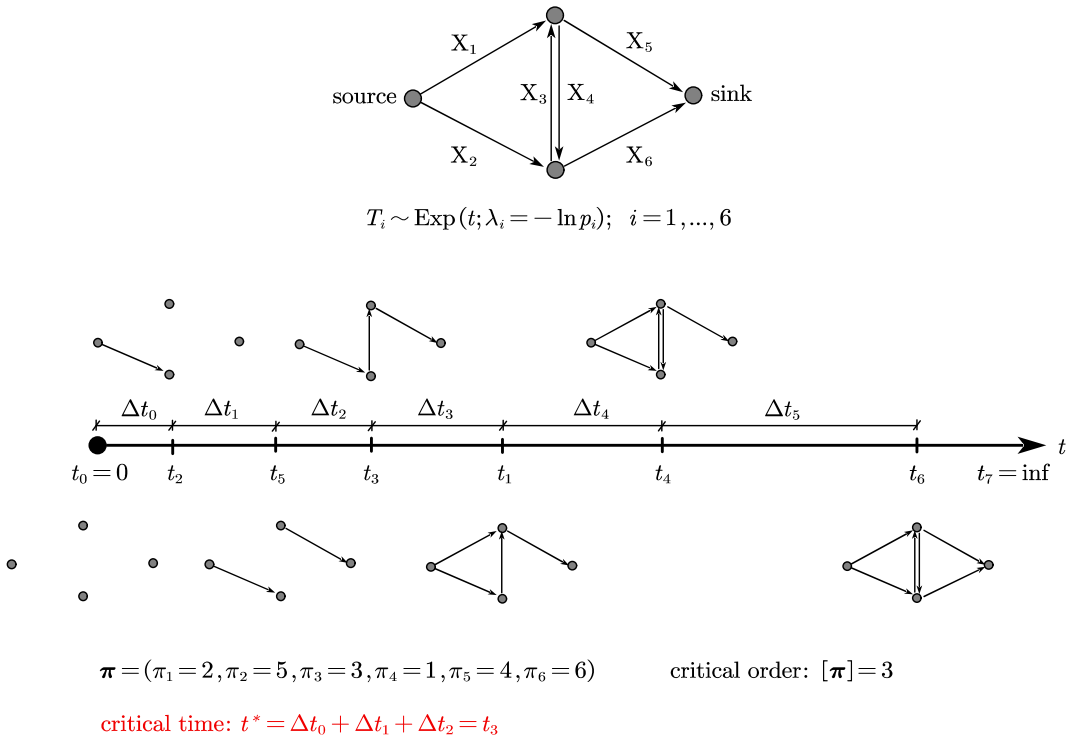


Figure 2.4: A realization of the creation process for the toy example. T_i denotes the exponential repair time of the i-th edge, where i ranges from 1 to 6. t_i denotes a realization (or an observed value) of T_i, and the collection of $t_1, \cdots, t_6$ defines a trajectory of the creation process, illustrated by the timeline in the figure. Given this trajectory, $\boldsymbol{\pi}, [\boldsymbol{\pi}], t^*, \Delta t_i$ represent the order in which the edges are constructed, the critical order, the critical time, and the sojourn time, respectively.

2.3.3.1.2 Permutation Monte Carlo

The creation process enables to devise an estimator of p_f that is often more efficient than crude MCS. Specifically, p_f can be rewritten as

$$p_f = \mathbb{E}_{\boldsymbol{\Pi}} \left[\mathbb{E}_{T^*|\boldsymbol{\Pi}} \left[\mathbb{I}\{T^* > 1\} \mid \boldsymbol{\Pi} \right] \right], \tag{2.24}$$

where the outer expectation can be estimated through a crude MCS estimator. That is:

$$\widehat{p}_f^{(pMCS)} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}_{T^*|\boldsymbol{\pi}^{(k)}} \left[\mathbb{I}\{T^* > 1\} \mid \boldsymbol{\Pi} = \boldsymbol{\pi}^{(k)} \right], \qquad \boldsymbol{\pi}^{(k)} \sim p_{\boldsymbol{\Pi}}(\boldsymbol{\pi}), k = 1, \cdots, N. \tag{2.25}$$

Here, N is the number of samples and $p_{\boldsymbol{\Pi}}(\boldsymbol{\pi})$, denotes the distribution of the permutation $\boldsymbol{\Pi}$. The estimator $\widehat{p}_f^{(pMCS)}$ is referred to as the pMCS estimator and is unbiased. Note that the idea of sampling in the permutation space can be traced back to at least [50]. However, the creation process provides a natural way of defining the permutation, that is, the order in which edges are repaired in the creation process. The probability of $\boldsymbol{\Pi} = \boldsymbol{\pi}$ can be calculated through [53]

$$\Pr(\boldsymbol{\Pi} = \boldsymbol{\pi}) = \int_0^{\infty} \int_{t_{\pi_1}}^{\infty} \cdots \int_{t_{\pi_{n-1}}}^{\infty} \prod_{i=1}^{n} \lambda_i \exp(-\lambda_i \cdot t_i) \mathrm{d}t_i = \prod_{i=1}^{n} \frac{\lambda_{\pi_i}}{\sum_{j=i}^{n} \lambda_{\pi_j}}. \tag{2.26}$$

Eq. (2.26) indicates a sequential sampling strategy of $\boldsymbol{\pi}$, wherein π_l is selected from indices that have not been previously chosen, that is, $\{j = 1, \cdots, n \mid j \neq \pi_1, \cdots j \neq \pi_{l-1}\}$, with probability proportional to λ_j.

In addition, given $\mathbf{\Pi} = \boldsymbol{\pi}$, the critical order $[\boldsymbol{\pi}]$ is fixed, and the sojourn time $\Delta T_l = T_{l+1} - T_l$ for $l = 1, \cdots, n-1$ follows the independent exponential distribution [17] which can be written as:

$$\begin{aligned}
\Pr(\Delta T_l > \tau, \mathbf{\Pi} = \boldsymbol{\pi}) &= \int_0^\infty \int_{t_{\pi_1}}^\infty \cdots \int_{t_{\pi_l+\tau}}^\infty \cdots \int_{t_{\pi_{n-1}}}^\infty \prod_{i=1}^{n} \lambda_i \exp(-\lambda_i \cdot t_i) \mathrm{d}t_i, \\
\Pr(\Delta T_l > \tau \mid \mathbf{\Pi} = \boldsymbol{\pi}) &= \frac{\Pr(\Delta T_l > \tau, \mathbf{\Pi} = \boldsymbol{\pi})}{\Pr(\mathbf{\Pi} = \boldsymbol{\pi})} = \exp(-\tau \sum_{j=l+1}^{n} \lambda_{\pi_j}).
\end{aligned} \tag{2.27}$$

Note that Eq. (2.27) also applies for $\Delta T_0 = T_1 - 0$ and is irrelevant to $\pi_{l+1}, \cdots, \pi_n$, i.e., $\Pr(\Delta T_l > \tau \mid \mathbf{\Pi} = \boldsymbol{\pi}) = \Pr(\Delta T_l > \tau \mid \pi_1, \cdots, \pi_l)$. This is because, when $\pi_1, \cdots, \pi_l$ are fixed, the summation $\pi_1, \cdots, \pi_l$ is also fixed. As it holds that $T^* = \sum_{l=0}^{[\boldsymbol{\pi}]-1} \Delta T_l$, the critical time T^* conditional on $\mathbf{\Pi} = \boldsymbol{\pi}$, denoted as $T^* \mid \boldsymbol{\pi}$, is the sum of $[\boldsymbol{\pi}]$ exponential distributed random variables, and follows a generalized Erlang distribution (or hypoexponential distribution). The CDF of the generalized Erlang distribution is well-known and can be evaluated through established algorithms. This facilitates computing the conditional expectation in Eq. (2.25), as it corresponds to the complement of the CDF of $T^* \mid \boldsymbol{\pi}^{(\boldsymbol{k})}$ evaluated at $t^* = 1$.

The pMCS method has a smaller variance than crude MCS [53] but involves more complex and costly computation when evaluating the expectation in Eq. (2.25), i.e., the generalized Erlang distribution's CDF. The computational cost of pMCS heavily depends on the algorithm used to calculate the CDF and the method for finding the critical number $[\boldsymbol{\pi}]$. Botev et al. [17] recommended employing the scaling and squaring algorithm [103] for evaluating the CDF of generalized Erlang distribution, which is more robust but less efficient than the algorithm utilized in Gertsbakh and Shpungin [64]. In cases where the exact calculation of the CDF is not feasible, employing approximation techniques is recommended [62]. Additionally, the determination of $[\boldsymbol{\pi}]$ requires multiple performance function evaluations. Parades et al. suggested a binary search method that requires evaluating the network performance at most $\log_2(n)$ times [114]. The efficiency of pMCS can be further enhanced by merging redundant edges during edge repairs [53], a technique also known as 'turnip' [64]. In connectivity-based problems, redundant edges are unrepaired edges with endpoints already connected by previously repaired edges since, when repaired, these edges will not form new connections. We refer to [53, 64] for a more comprehensive and detailed introduction to the 'turnip' technique.

In general, the pMCS method is unsuitable for rare event estimation, e.g., when the network is highly reliable, and p_f is small. For such cases, the creation process needs to be combined with the multilevel splitting [104, 17, 18] or importance sampling [71]. This will be detailed in Subsection 2.3.4.4. Also, note that, besides the network connectivity, pMCS applies to any coherent network performance function. So far, the input random variables must be independent and Bernoulli distributed, but this restriction will be relaxed in the following section.

2.3.3.2 Creation process for systems with multi-state inputs

The creation process can be extended to encompass multi-state systems [63, 18, 24]. Let $\boldsymbol{X} = (X_1, \cdots, X_n)$ denote the input random vector that collects the capacity of all edges, and $g^{(\mathrm{mf})}(\boldsymbol{x})$ be the network performance function. Suppose each input variable X_d for $d = 1, \cdots, n$ is independent and can take $n_d + 1$ states, denoted as $s_{d,0} < s_{d,1} < \cdots < s_{d,n_d}$, with probability $p_{d,0}, p_{d,1}, \cdots, p_{d,n_d}$, respectively. The objective is then to estimate the failure probability defined as $p_f \triangleq \Pr(g^{(\mathrm{mf})}(\boldsymbol{X}) < \gamma)$.

In the literature, multiple strategies have been developed for extending the creation process to multi-state systems [63, 18, 24, 64]. One possibility is to associate each state $s_{d,i}$, for $i = 1, \cdots, n_d$ and $d = 1, \cdots, n$, with an independent exponentially distributed random variable $T_{d,i}$ with rate $\lambda_{d,i}$ [18]. Each d-th edge's capacity is initially set to its minimal value, $s_{d,0}$, and progressively restored as the artificial time t reaches the values in $T_{d,1}, \cdots, T_{d,n_d}$. In particular, at time $t = T_{d,i}, i = 1, \cdots, n_d$, if the current edge capacity X_d is less than $s_{d,i}$, it is upgraded to $s_{d,i}$; otherwise, no action is taken. We therefore interpret $T_{d,i}$ as the deadline for restoring the d-th edge, X_d, to at least its i-th state $s_{d,i}$. In this way, an extended creation process denoted as $\boldsymbol{X}(t) = (X_1(t), \cdots, X_n(t))$ can be defined. For $d = 1, \cdots, n$ and $i = 0, \cdots n_d - 1$, $X_d(t) \leq s_{d,i}$ holds if and only if the deadlines for restoring X_d to larger capacities are not met, i.e., $T_{d,j} > t$ for $j = i + 1, \cdots, n_d$. Hence, it holds that $\Pr(X_d(t) \leq s_{d,i}) = \exp(-t \sum_{j=i+1}^{n_d} \lambda_{d,j})$. The rate parameters $\lambda_{d,i}, i = 1, \cdots, n_d, d = 1, \cdots, n$ are then chosen such that the marginal distribution of the process at time $t = 1$, $\boldsymbol{X}(1)$, coincides with the initial static model. In other words, for each $d = 1, \cdots, n$, it must hold that:

$$\Pr(X_d(1) \leq s_{d,i}) = \exp(-\sum_{j=i+1}^{n_d} \lambda_{d,j}) = \sum_{j=0}^{i} p_{d,j}, \quad i = 0, \cdots n_d - 1. \tag{2.28}$$

By solving the equation set in Eq. (2.28), it can be derived that

$$\begin{aligned} \lambda_{d,n_d} &= -\ln(\sum_{j=0}^{n_d-1} p_{d,j}) = -\ln(1 - p_{d,n_d}), \\ \lambda_{d,i} &= -\ln(\sum_{j=0}^{i-1} p_{d,j}) - \lambda_{d,i+1} - \cdots - \lambda_{d,n_d}, \quad i = n_d - 1, \cdots, 1. \end{aligned} \tag{2.29}$$

Now, $\Pr(g^{(\mathrm{mf})}(\boldsymbol{X}) < \gamma)$ can be written as $\Pr(g^{(\mathrm{mf})}(\boldsymbol{X}(1)) < \gamma)$. Moreover, since the performance function $g^{(\mathrm{mf})}$ is coherent (or monotonic), $\Pr(\phi(\boldsymbol{X}) \leq \gamma)$ is equivalently the probability that the critical time T^*, when $g^{(\mathrm{mf})}$ first reaches γ, is greater than $t = 1$, i.e., $\Pr(g^{(\mathrm{mf})}(\boldsymbol{X}) < \gamma) = \Pr(T^* > 1)$. The critical time T^* can be expressed as a function of all $T_{d,i}$'s, a total of $\sum_{d=1}^{n} n_d$ independent exponentially distributed variables. After collecting these variables in a larger vector, i.e., letting $\boldsymbol{T} \triangleq (T_{1,1}, \cdots, T_{1,n_1}, \cdots, T_{n,1}, \cdots, T_{n,n_n})$, a pMCS estimator similar to Eq. (2.25) can be defined for multi-state systems. The permutation, however, is now defined over $\sum_{d=1}^{n} n_d$ states (edge capacities) in $\boldsymbol{T}$.

An alternative approach is the multi-level creation process [24]. In this process, each edge indexed by d is linked to an exponentially distributed repair time, T_d, with rate λ_d. The edge is initially set to its worst state, $s_{d,0}$, and undergoes an independent repair at time T_d. Following this repair, the new state is determined by the time interval during which the repair oc-

curs. Specifically, suppose that the timeline $[0, \infty)$ is partitioned into the following $n_d + 1$ subintervals: $[\tau_{d,0} = 0, \tau_{d,1}), [\tau_{d,1}, \tau_{d,2}), \cdots, [\tau_{d,n_d-1}, \tau_{d,n_d} = 1)$, and $[\tau_{d,n_d} = 1, \tau_{d,n_d+1} = \infty)$, with $0 < \tau_{d,1} < \tau_{d,2} \cdots < \tau_{d,n_d-1} < 1$ denoting $n_d - 1$ distinct levels. In the multi-level creation process, the post-repair state of the d-th edge is $s_{d,i}$ if and only if the repair takes place during the $(n_d + 1 - i)$-th interval, i.e., $\tau_{n_d-i} \leq T_d < \tau_{n_d+1-i}$. Recall that $s_{d,0} < \cdots < s_{d,n_d}$. Thus, an earlier repair will never degenerate the edge's state. In addition, to be consistent with the static model, the repair rate λ_d and the levels $\tau_1, \cdots, \tau_{n_d-1}$ must be chosen such that $\Pr(X_d(t = 1) = s_{d,i}) = p_{d,i}$ for each $d = 1, \cdots, n$ and $i = 0, \cdots, n_d$. Here, X_d denotes the state for the d-th edge and is a function of the artificial time t in the multi-level creation process. Consequently, it can be derived that $\lambda_d = -\ln(p_{d,0})$, and $\tau_{d,i} = \frac{\ln(1-p_{d,n_d}-\cdots-p_{d,n_d-i+1})}{\ln(p_{d,0})}$, for $d = 1, \cdots, n$ and $i = 1, \cdots, n_d - 1$. It is possible to devise a pMCS method with the multi-level creation process; however, given a permutation $\boldsymbol{\pi}$, the critical time $T^* = \sum_{l=0}^{[\boldsymbol{\Pi}]-1} \Delta T_l$ no longer follows a generalized Erlang distribution. This is because the critical order $[\boldsymbol{\pi}]$ given $\boldsymbol{\pi}$ is a random variable that also depends on the sojourn time $\Delta T_l, l = 1, \cdots, n$. In general, large sojourn time leads to poor post-repair performance and, hence, a potentially large critical order.

2.3.3.3 Smoothing the discrete network performance with the creation process

In the creation process, one estimates $\Pr(T^* > 1)$ instead of $\Pr(g^{(\text{conn})}(\boldsymbol{X}) = 0)$ or $\Pr(g^{(\text{mf})}(\boldsymbol{X}) < \gamma)$, where the critical time T^* is a deterministic function of a set of independent exponential distributions, i.e., $T_1, \cdots, T_n$, and is a continuous random variable. In this way, the original network performance g, which is discrete, is substituted with a continuous one, that is, T^*. To illustrate the smoothing effect of the creation process, let us consider a system with only one binary-state component X, which can either fail, denoted as $X = 0$, or not fail, denoted as $X = 1$, with probability $p = 0.001$ and $1 - p$, respectively. The network performance is defined as $g(X) = X$, and the probability $\Pr(g(X) = X = 0)$ is of interest. The CDF of $g(X)$, or equivalently X, is a discontinuous step function with a 'jump' of p at 0 and $1-p$ at 1. The creation process of X is governed by an artificial repair time T, which follows the exponential distribution with rate $\lambda = -\ln p$. Since the system only has one component, the critical time T^* is identical to T and has a continuous exponential CDF. The difference between the CDF of $g(X) = X$ and T^* is further illustrated in Fig. 2.5.

The new performance function $T^*(t_1, \cdots, t_n)$ facilitates leveraging rare event estimation techniques originally proposed for continuous system performance to solve network reliability problems, but it is also more expensive to compute, often requiring multiple calls to the original performance function g. For tackling connectivity-based problems, the creation process has been incorporated in multi-level splitting [104], generalized splitting [17], sequential Monte Carlo [133], importance sampling [71], among others. For maximum flow problems, Botev et al., [18] combined the modified creation process with the generalized splitting method, and Cancela et al., [24] integrated their multi-level creation process with splitting. These combinations will be detailed later in Subsection 2.3.4.4.

2.3.4 Multi-level splitting methods

Multi-level splitting is a popular technique for estimating rare events. The idea of splitting dates back to [79] and has been intensively investigated in the past three decades. Many variants have been

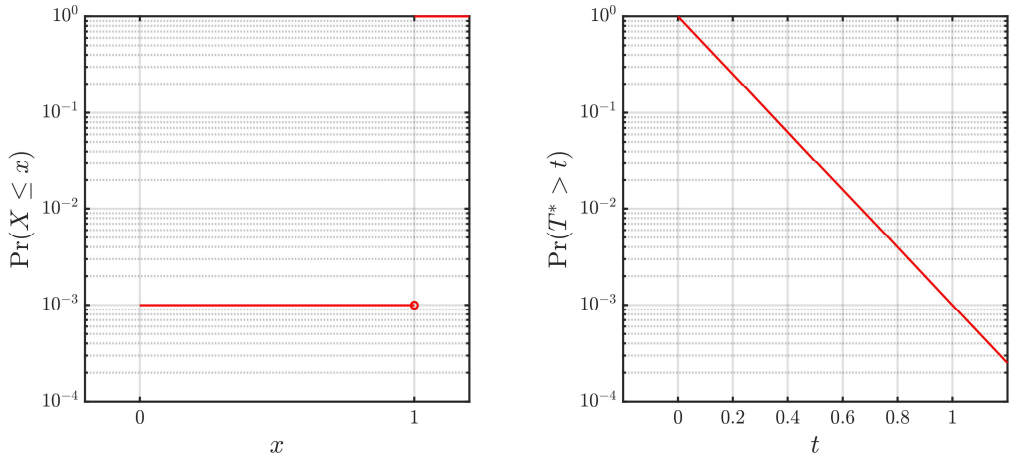


Figure 2.5: Left: the CDF of the $g(X) = X$, which is a Bernoulli distributed variable, is also depicted in the red line. Right: the probability that $T^* > t$ of the example in Subsection 2.3.3.3.

developed in different contexts, sometimes with different names, e.g., SuS in civil engineering [7], the RESTART method in telecommunication [135], nested sampling within the context of Bayesian analysis [129]. The core idea, however, is similar. That is to express the rare event, denoted as F, as the intersection of nested intermediate events $F_0 \subset F_1 \subset \cdots \subset F_M = F$, i.e.,

$$F = \bigcap_{l=0}^{M} F_l, \tag{2.30}$$

and to decompose the probability of the rare event as the product of a sequence of larger conditional probabilities, i.e.,

$$p_f = \Pr(F) = \prod_{l=1}^{M} \Pr(F_l | F_{l-1}). \tag{2.31}$$

Usually, $F_0 = \Omega_{\boldsymbol{X}}$ represents the initial sample space, so its probability measure equals one. While $\Pr(F_1|F_0)$ can be directly estimated through crude MCS, the remaining conditional probabilities are estimated by splitting the previous trajectories. The final estimator takes the form:

$$\widehat{p}_f^{(MS)} = \prod_{l=1}^{M} \widehat{\Pr}\left(F_l \mid F_{l-1}\right). \tag{2.32}$$

A comprehensive review of this group of methods and their applications in different fields is out of the scope of this thesis. Instead, we aim to illustrate the basic idea and major variants in a general mathematical framework known as the *overflow problem* [57, 27], where the rare event is modeled as the threshold exceedance by a deterministic function. Building on this preliminary knowledge, we summarize recent developments in multi-level splitting techniques for network reliability assessment.

2.3.4.1 Problem settings

Let $\boldsymbol{Y}(t)$ denote a time-homogenous Markov process with a state space $\boldsymbol{y} \in \Omega_{\boldsymbol{Y}}$, a fixed initial point $\boldsymbol{y}(0)$, and a transitional kernel $K(\boldsymbol{y} \mid \boldsymbol{y}^*)$ that represents the conditional distribution of $\boldsymbol{y}$

given $\boldsymbol{y}^*$. Events $\mathcal{A}$ and $\mathcal{B}$ are two disjoint subsets of $\Omega_{\boldsymbol{Y}}$, characterized by a deterministic function $h : \Omega_{\boldsymbol{Y}} \to \mathbb{R}$ and a threshold γ. The objective is then to estimate the probability that the random process reaches $\mathcal{B}$ before $\mathcal{A}$. This is referred to as the dynamic overflow problem. In particular, define $\mathcal{A} \triangleq \{\boldsymbol{y} \in \Omega_{\boldsymbol{Y}} | h(\boldsymbol{y}) < 0\}$ and $\mathcal{B} \triangleq \{\boldsymbol{y} \in \Omega_{\boldsymbol{Y}} | h(\boldsymbol{y}) \geq \gamma > 0\}$, and let $\tau_{\mathcal{A}}$ and $\tau_{\mathcal{B}}$ denote the first time $\boldsymbol{Y}(t)$ enters $\mathcal{A}$ and $\mathcal{B}$, respectively. The target probability can be written as $\Pr(\tau_{\mathcal{B}} \leq \tau_{\mathcal{A}})$.

Alternatively, one substitutes the Markov process $\boldsymbol{Y}(t)$ with a d-dimensional vector $\boldsymbol{X}$ and estimates the probability that a deterministic function g of $\boldsymbol{X} \in \Omega_{\boldsymbol{X}}$ is greater or equal to a specified threshold γ, i.e., $\Pr(g(\boldsymbol{X}) \geq \gamma)$. Let $\Omega_{\boldsymbol{X}}$ denote the sample space of $\boldsymbol{X}$, and $p_{\boldsymbol{X}}(\boldsymbol{x})$ be its distribution. This is the static overflow problem. Note that all network performance functions discussed in this thesis, i.e., $g^{(\text{conn})}, g^{(\text{mf})}$, and $g^{(\text{bz})}$, can be formulated as static overflow problems. In particular, the failure probability $\Pr(g^{(\text{conn})} = 0)$ equals $\Pr(g^{(\text{conn})} \geq 1)$, and $\Pr(g^{(\text{mf})} < \gamma)$ can be equivalently expressed as $\Pr(-g^{(\text{mf})} \geq -\gamma) + \epsilon$, where $\epsilon > 0$ denotes a negligible positive number, e.g., 10^{-20}.

The dynamic overflow problem can be reformulated as a static one by interpreting $\Omega_{\boldsymbol{X}}$ as a functional space of trajectories (or realizations) of $\boldsymbol{Y}(t)$, denoted as $\boldsymbol{y}(t)$. The performance function $g(\boldsymbol{y}(t)) \triangleq \sup_{0 \leq t \leq \tau_{\mathcal{A}}} h(\boldsymbol{y}(t))$ then maps each trajectory $\boldsymbol{y}(t), t > 0$ to its maximum h value before entering $\mathcal{A}$. On the other hand, by devising an artificial Markov chain with an augmented state space, the static overflow problem can also be reformulated in a dynamic setting [26, 19].

When the threshold γ is large, $\Pr(\tau_{\mathcal{B}} \leq \tau_{\mathcal{A}})$ (or $\Pr(g(\boldsymbol{X}) \geq \gamma)$) is often small, and crude MCS becomes impractical when simulating $\boldsymbol{Y}(t)$ and computing the deterministic function h (or g) are computationally demanding. In fact, to maintain a constant coefficient of variation in MCS, the number of trajectories to sample is proportional to $\frac{1}{\Pr(\tau_{\mathcal{B}} \leq \tau_{\mathcal{A}})}$ (or $\frac{1}{\Pr(g(\boldsymbol{X}) \geq \gamma)}$). Multi-level splitting is one of several variance reduction techniques that address this issue.

2.3.4.2 Implementation details

2.3.4.2.1 Splitting for dynamic overflow problems

In dynamic overflow problems, the intermediate events $\{F_l\}_{l=0}^{M}$ are defined as follows:

$$\begin{aligned} \mathcal{B}_l &\triangleq \{\boldsymbol{y} \in \Omega_{\boldsymbol{Y}} \mid h(\boldsymbol{y}) \geq \gamma_l\}, \\ F_l &\triangleq \{\boldsymbol{y}(t) \in \Omega_{\boldsymbol{Y}(t)} \mid \tau_{\mathcal{B}_l} \leq \tau_{\mathcal{A}}\}, \end{aligned} \tag{2.33}$$

where $-\inf = \gamma_0 < \gamma_1 < \cdots < \gamma_M = \gamma$ denotes a sequence of levels that characterizes the intermediate events. M is the number of intermediate levels, and $\Omega_{\boldsymbol{Y}(t)}$ denotes the trajectory space of the process $\boldsymbol{Y}(t)$. Since $\mathcal{B}_{l-1} \subset \mathcal{B}_l$ for each $1 \leq l \leq M$, it is impossible to enter $\mathcal{B}_{l-1}$ later than $\mathcal{B}_l$; otherwise put, $\tau_{\mathcal{B}_{l-1}} \leq \tau_{\mathcal{B}_l}$. Consequently, F_l is nested in F_{l-1} for $1 < l < M$, and both Eqs. (2.31) and (2.30) are valid.

For efficiently estimating the conditional probabilities, the multi-level splitting methods begin with N_0 independent trajectories (or realizations) of the Markov process $\boldsymbol{Y}(t)$. Each trajectory is simulated until it enters either $\mathcal{B}_1$ or $\mathcal{A}$. Trajectories entering the event $\mathcal{A}$ first are terminated immediately, while the remaining ones are split at the moment they reach $\mathcal{B}_1$, denoted as $\tau_{\mathcal{B}_1}$. Specifically, each k-th survival trajectory is replicated $n_1^{(k)}$ times until $\tau_{\mathcal{B}_1}$. Assume there are in total R_1 survival

trajectories. The total number of copies (or offsprings), denoted as N_1, is then given by $\sum_{k=1}^{R_1} n_1^{(k)}$, and the ratio $\frac{R_1}{N_0}$ is an unbiased estimator of $\Pr(F_1 \mid F_0)$. In the subsequent iteration, each trajectory copy is extended independently from its entrance state until it enters either $\mathcal{B}_2$ or $\mathcal{A}$. Note that these trajectories are not independent since some of them share parts of history. The number of survival trajectories and their copies are then denoted as R_2 and $N_2 = \sum_{k=1}^{R_2} n_2^{(k)}$, respectively, and the ratio $\frac{R_2}{N_1}$ is employed as an unbiased estimator for $\Pr(F_2 \mid F_1)$. The procedure continues until R_M trajectories reach the rare event $F_M = F$. The final estimator of $\Pr(F)$ is then given by

$$\widehat{p}_f^{(MS)} = \prod_{l=1}^{M} \widehat{\Pr}\left(F_l \mid F_{l-1}\right) = \prod_{l=1}^{M} \frac{R_l}{N_{l-1}}. \tag{2.34}$$

There are two different ways of splitting: fixed splitting (FS) and fixed efforts splitting (FE) [57]. In FS, each survival trajectory is replicated a fixed number η_l of times at the l-th iteration, i.e., $n_l^{(k)} = \eta_l$ for each k, resulting in a total of $N_l = \eta_l \cdot R_1$ copies (or offspring). The parameter η_l is also known as the splitting factor in literature [57, 114]. The optimal but impractical choice of η_l is $\frac{1}{\Pr(F_{l+1}|F_l)}$ [65]. The choice of a large η_l leads to an explosion of N_l, while a small η_l may cause the extinction of the trajectories. In both cases, the efficiency of FS is poor. By contrast, in FE, N_l is fixed and needs to be assigned to R_l survival trajectories. For instance, one can generate N_l copies uniformly with replacement from R_l survival trajectories. This is known as random assignment in literature [57, 59, 85]. The number of replicas for each survival trajectory then follows the multinomial distribution. Alternatively, one can replicate each survival trajectory $\lfloor \frac{N_l}{R_l} \rfloor$ times, and sample uniformly without replacement the remaining $N_l - R_l \cdot \lfloor \frac{N_l}{R_l} \rfloor$ copies. This is known as the fixed assignment. Other assignment methods include residual sampling [26] and stratified sampling [85].

Regardless of whether FS or FE is utilized and irrespective of the assignment method in FE, the multi-level splitting estimator (see Eq. 2.34) is unbiased [85]. However, they differ in the estimator variance. Garvels and Kroese [57] concluded in their comparative study that FE with fixed assignment usually leads to smaller variance. In some simplified settings (e.g., when M=2), an analytical expression of the variance can be derived by analyzing the entrance states [58].

The efficiency of the multi-level splitting method relies on an appropriate choice of intermediate levels $\{\gamma_l\}_{l=1}^{M}$, which are often specified through a pilot run. Although $\mathcal{A}$ and $\mathcal{B}$ are defined through the deterministic function h, any other function $\widetilde{h} : \Omega_{\boldsymbol{Y}} \to \mathbb{R}$ such that $\mathcal{A} \triangleq \{\boldsymbol{y} \in \Omega_{\boldsymbol{Y}} \mid \widetilde{h}(\boldsymbol{y}) < \widetilde{\gamma}_0\}$ and $\mathcal{B} \triangleq \{\boldsymbol{y} \in \Omega_{\boldsymbol{Y}} \mid \widetilde{h}(\boldsymbol{y}) \geq \widetilde{\gamma}\}$, can be employed to set levels and to drive the Markov process. The selection of $\widetilde{h}$, however, is non-trivial and still remains an open question. Additional insights are given in the discussion on the optimal importance function in Garvels et al. [58].

2.3.4.2.2 Generalized splitting for static overflow problems

Splitting Multi-level splitting can be adapted to static overflow problems [15]. Thereby, the intermediate events are defined by:

$$F_l \triangleq \{\boldsymbol{X} \in \Omega_{\boldsymbol{X}} \mid g(\boldsymbol{x}) \geq \gamma_l\}. \tag{2.35}$$

Here, $-\inf = \gamma_0 < \gamma_1 < \cdots < \gamma_M$ denotes the sequence of intermediate levels. It is evident that these intermediate events are nested, so $\Pr(F)$ can be estimated by Eq. (2.32). For estimating

the conditional probabilities $\{\Pr(F_l \mid F_{l-1})\}_{l=1}^{M}$ in static settings, the multi-level splitting method proceeds as follows. At the initial iteration, the method generates N_0 independent samples from $p_{\boldsymbol{X}}(\boldsymbol{x})$, removes those falling outside of F_1, and replicates samples that remain. These survival samples are also known as the seeds or elite samples. Suppose there are R_1 seeds, and after either fixed splitting or fixed effort splitting, each k-th seed is replicated $n_1^{(k)}$ times. The total number of replicas N_1 then equals $\sum_{k=1}^{R_1} n_1^{(k)}$. These N_1 replicas follow the distribution $p(\boldsymbol{x} \mid F_1)$, but are highly correlated. To reduce the dependence among replicas, a transition kernel K_1 that is invariant to $p(\boldsymbol{x} \mid F_1)$ is applied to each replica, resulting in N_1 enriched samples that still follow $p(\boldsymbol{x} \mid F_1)$. This is equivalent to arranging N_1 Markov kernels in R_1 'forks', with $n_1^{(k)}$ branches rooting in the k-th seed. Suppose R_2 out of N_1 samples fall in F_2. The above splitting and one-step Markov move are then iterated until $l = M$, and Eq. (2.34) is an unbiased estimator of $\Pr(F)$.

The multi-level splitting methods can also be interpreted as the particle integration method with the Feynman-Kac model [26, 114]. Hence, various theoretical results available for particle integration methods can be leveraged to establish the unbiasedness, asymptotic variance, and other properties of the above multi-level splitting estimator [43].

Generalized splitting To further reduce the dependence among samples, it is possible to organize Markov kernels into multiple chains rather than forks, thus allowing them to mix properly. This approach is referred to as the generalized splitting in Botev and Kroese [16]. Specifically, let us consider generating n samples from the distribution $p(\boldsymbol{x} \mid F_l)$ with a seed $\boldsymbol{x}^*$ in F_l. In splitting, this is done by sampling from the transition kernel $K_l(\boldsymbol{x} \mid \boldsymbol{x}^*)$, i.e., $\boldsymbol{x}_j \sim K_l(\boldsymbol{x} \mid \boldsymbol{x}^*)$ for all $j = 1, \cdots, n$. Recall that $K_l(\boldsymbol{x} \mid \boldsymbol{x}^*)$ is the conditional probability (or density) of $\boldsymbol{x}$ given the seed $\boldsymbol{x}^*$ and is invariant to $p(\boldsymbol{x} \mid F_l)$. Since these samples share the same parent, they can be highly correlated. By contrast, in generalized splitting, the j-th sample $\boldsymbol{x}_j$ is generated from $K_l(\boldsymbol{x} \mid \boldsymbol{x}_{j-1})$ with $j = 1, \cdots, n$ and $\boldsymbol{x}_0 = \boldsymbol{x}^*$, thus decreasing the dependence between, e.g., $\boldsymbol{x}_1$ and $\boldsymbol{x}_n$. At each level of splitting, the transition kernels are determined using Markov Chain Monte Carlo (MCMC) algorithms. Selecting an inappropriate algorithm can result in highly correlated intermediate samples, leading to significant errors. In the context of generalized splitting, commonly used MCMC algorithms include adaptive conditional sampling [111] (an adaptive variant of the preconditioned Crank-Nicolson algorithm [41]), Hamiltonian Monte Carlo [106], Gibbs sampling [60], among others.

The generalized splitting can be directly embedded into the multi-level splitting method by replacing the original splitting and one-step Markov move. The initial samples are still independently drawn from $p_{\boldsymbol{X}}(\boldsymbol{x})$, but in the subsequent iterations, samples are generated by initiating a Markov chain from each seed. Suppose that, at each l-th iteration, R_l out of N_{l-1} samples fall in F_l. The final estimator of $\Pr(F)$ is then given by Eq. (2.34).

In Botev and Kroese [16], the authors devised a fixed generalized splitting scheme, where the average length of the chain at iteration l equals $\frac{1}{\widehat{\Pr}(F_l|F_{l-1})}$, with $\widehat{\Pr}(F_l \mid F_{l-1})$ estimated from a pilot run. This fixed generalized splitting method is proven to be unbiased. On the other hand, one can also devise a fixed effort generalized splitting scheme in which the total number of samples is fixed at each iteration. However, since now the length of the chain depends on the sampling history, in particular, the number of seeds at each iteration, the proof in Botev and Kroese [16] no longer applies, and the unbiasedness of this method needs to be revalidated.

2.3.4.2.3 The adaptive selection of the levels and subset simulation

The efficiency of the multi-level splitting (or generalized splitting) algorithms is heavily dependent on the choice of the intermediate events or, equivalently, on how the levels are structured. However, an appropriate selection of levels requires a comprehensive prior knowledge of the system, which is often not available before the simulation. Most of the aforementioned multi-level splitting methods require a pilot run to determine the levels. However, obtaining a good choice of levels necessitates a substantial pilot cost, ultimately reducing the overall efficiency, and it can be difficult to balance the cost in advance.

The intermediate levels can also be chosen adaptively during the splitting algorithm, such that the conditional probability $\widehat{\Pr}(F_l \mid F_{l-1})$ is approximately equal to a constant value p_0. This is known as the adaptive multi-level splitting (AMS) method. This approach was first discussed in Garvels [59] for solving dynamic overflow problems and was subsequently investigated in Cerou and Guyader [27] for a one-dimensional Markov process. In the approach, at the l-th iteration of AMS, the level γ_l is selected as an empirical p_0-quantile of the maximal h values reached by the N_{l-1} trajectories generated in the previous iteration. Only the trajectory with maximal h value larger or equal to γ_l is kept and further replicated until the moment it first enters $\mathcal{B}_l \triangleq \{\boldsymbol{y} \in \Omega_{\boldsymbol{Y}} \mid h(\boldsymbol{y}) \geq \gamma_l\}$. Suppose R_l trajectories survive and split into N_l replicas through either fixed or fixed effort splitting. These replicas are then simulated until the end, i.e., until they first enter $\mathcal{A}$, to determine the next level γ_{l+1}. An interesting variant of AMS is to choose γ_l as the smallest maximal h value among trajectories, which is often referred to as the 'last sample' implementation. This variant can be viewed as a limiting case of the fixed-effort multi-splitting method, where the number of iterations M approaches infinity, so it is also unbiased [29].

Botev and Kroese [15] further embedded the generalized splitting in AMS and proposed the adaptive generalized splitting method (ADAM). Although ADAM is only used as a pilot run for the generalized splitting method in Botev and Kroese [16], it can be employed as a stand-alone algorithm for estimating $\Pr(F)$. In Chapter 7 of this thesis, we show that ADAM is even more efficient than generalized splitting when taking into account the additional pilot cost for the latter. Note that the ADAM method was initially and independently proposed in the field of civil engineering, where it is referred to as SuS [7]. Both SuS and ADAM are asymptotically unbiased [7].

When $g(\boldsymbol{X})$ is a continuous random variable, and samples can be generated independently from $p\left(\boldsymbol{x} \mid g(\boldsymbol{X}) \geq \gamma_l\right)$, Cerou et al. [26] proved that this ideal version of AMS is only asymptotically unbiased and derived its limiting distribution as the sample size $N_l, l = 1, \cdots, M$ becomes infinity. Interestingly, the 'last sample' variant of AMS is unbiased in this ideal case [67].

A brief summary of the aforementioned multi-level methods is given in Table. 2.2. Note that whether a multi-level splitting method is unbiased or not is sensitive to slight changes in the algorithm and the problem. For instance, the unbiasedness of the algorithm proposed by Guyader et al. [67] builds on the continuity of the random variable $g(\boldsymbol{X})$, and the generalized splitting method in Botev and Kroese [16] may not maintain its unbiasedness when employing a fixed effort scheme.

Table 2.2: Different multi-level splitting methods

reference	problems	levels	splitting approach	unbiased
main Alg. [59, 85]	dynamic	fixed	FS,FE+RA,FE+FA	yes
AMS [29, 27]	dynamic	adaptive	FE+RS	unknown
main Alg. [28]	dynamic	adaptive	FE+RS	yes
SuS+splitting [37]	dynamic	adaptive	FS	asymptotically
Alg.1 [26]	static	fixed	FE+RS	yes
Alg.2 [26]	static	adaptive	ideal	asymptotically
main Alg. [67]	static	adaptive	ideal	yes
Alg.3.1 [15]	static	fixed	FE+RA	yes
Alg.3.2 [15]	static	adaptive	FE+RA	unknown
GS [16]	static	fixed	GFS	yes
ADAM [16]	static	adaptive	GFE+FA	asymptotically
Alg.5.1 [16]	static	fixed	FE+FA	yes
SuS [7]	static	adaptive	GFE+FA	asymptotically

Abbreviations: Alg: algorithm; AMS: adaptive multi-level splitting; GS: generalized splitting; ADAM: adaptive generalized splitting; SuS: subset simulation; dynamic: dynamic overflow problems; static: static overflow problems; FS: fixed splitting; GFS: generalized fixed splitting; FE+FA: fixed effort splitting with fixed assignment; FE+RA: fixed effort splitting with random assignment; FE+RS: fixed effort splitting with residual sampling; GFE+FA: generalized fixed effort splitting with fixed assignment; ideal: direct sample from $p(\boldsymbol{x} \mid F_l)$

2.3.4.3 The optimal selection of parameters in an ideal case

In practice, the selection of the parameters in the multi-level splitting methods is of great importance. A bad choice will decrease the efficiency of the algorithm. The optimal selection of parameters can be derived for an ideal but impractical case, where samples are generated independently from $p(\boldsymbol{x} \mid F_{l-1})$ for estimating the conditional probability $\Pr(F_l \mid F_{l-1})$ [57]. In such a case, Eq. (2.32) becomes a product of a sequence of crude MCS estimators.

Regarding fixed-effort splitting, Garvels and Kroese [57] demonstrated that, given a substantial computational budget (e.g., the overall number of samples or trajectories generated by the algorithm), the variance of the estimator in Eq. (2.32) can be minimized by configuring parameters as follows: set $M \approx -\log(\Pr(F))/2$, $N_0 = N_1 = \ldots = N_{M-1}$, and choose levels such that $\Pr(F_l|F_{l-1}) = \exp(-2) \approx 0.135$. Recall that M is the number of levels, and $N_l, l = 0, \cdots, M-1$ denotes the sample size at each iteration.

Using the multi-level branching theory, Glasserman et al. [65] derived a similar strategy for selecting fixed splitting parameters. In particular, they suggest setting $M \approx -\log(\Pr(F))/2, s_1 = s_2, \cdots, = s_{M-1} \approx e^2$ and selecting levels such that $\Pr(F_l|F_{l-1}) = \exp(-2) \approx 0.135$. Recall that η_l denotes the splitting factor for the l-th level. The initial sample size N_0 should be equal to $\frac{1}{M}$ of the total budget.

These theoretical results derived for the ideal case shed insights into choosing parameters in practical settings. For instance, when adaptively choosing the levels in AMS, ADAM, or SuS, the conditional probabilities should be equal, and the sample size at each iteration should also be equal. However, all of these results are based on the implicit assumption that, given the $(l-1)$-th level γ_{l-1}, there

is always a $\gamma_l > 0$ such that $\Pr(F_l \mid F_{l-1}) = \exp(-2)$. As will be shown later in this thesis, this assumption does not always hold in network reliability assessment, particularly when $g(\boldsymbol{X})$ is discrete, with significant 'jumps' in its CDF. In such cases, it is advisable to adapt the conditional probabilities and the sample size per level. This further motivates the idea of the adaptive effort subset simulation method (aE-SuS) in Chapter 4.

2.3.4.4 Multi-level splitting methods for network reliability assessment

In network reliability assessment, most standard problems can be reduced to static overflow problems, as described in Subsection 2.3.4.1. Different from structural reliability assessment, the network performance $g(\boldsymbol{X})$ is often a discrete random variable due to binary or multi-state variables in $\boldsymbol{X}$ or due to the discrete nature of g. By contrast, all adaptive methods listed in Table 2.2 for static problems, along with their corresponding mathematical analyses, are tailored to continuous performance, potentially limiting their applicability to discrete performance. For instance, the 'last particle' version of the AMS method cannot be used [67], and SuS will lead to substantial errors when significant 'jumps' appear in the CDF of $g(\boldsymbol{X})$ [7]. At first glance, fixed-level methods may not appear to suffer from this discontinuity issue. However, they heavily depend on pilot runs, often utilizing adaptive methods, and therefore, still face potential issues. In the literature, multi-level splitting methods are commonly integrated with the creation process for rare event estimation in network analysis [38, 17, 104, 18, 24, 25], but this is not always necessary. In fact, we find that it is often more efficient to directly adapt the conditional probabilities and sample size in SuS, and this novel approach will be detailed in Chapter 4. Note that in certain situations the network performance is continuous or approximately continuous, especially in large-scale networks, and in such cases, SuS is directly applicable [143, 144, 77].

The creation process can be integrated into the multi-level splitting. In the creation process, the network performance $g(\boldsymbol{X})$ is replaced by the critical time T^* when the network first becomes functional, and one estimates the probability of T^* exceeding 1, instead of $\Pr(g(\boldsymbol{X}) \geq \gamma)$. On the one hand, this is still a static overflow problem, but with a continuous performance $T^*(T_1, \cdots, T_n)$, which can be directly integrated with not only GS [17, 18], but also any other multi-level splitting methods for estimating rare events in networks. On the other hand, the creation process with exponential repair time is a Markov process [17] and provides a natural way of defining the dynamic overflow problem. The graph-evolution trajectory $\widetilde{\mathcal{G}}(t)$ terminates when the graph first becomes functional at the critical time T^*, i.e., $\tau_{\mathcal{A}} \triangleq T^*$, and $\tau_{\mathcal{B}} \triangleq 1$. It is then evident that $\Pr(T^* > 1)$ can be reformulated as a dynamic overflow problem $\Pr(\tau_{\mathcal{B}} < \tau_{\mathcal{A}})$. The intermediate levels for splitting are positioned along the artificial timeline [104, 24], denoted as $0 = \tau_0 < \tau_1 < \cdots < \tau_M = 1$, and the intermediate domains are defined by $\{\widetilde{\mathcal{G}}(t) \mid \tau_l < \tau_{\mathcal{A}}\}$.

Alternatively, multi-level splitting can be incorporated into pMCS methods for generating pivotal permutations with a large critical order [133]. These permutations often correspond to a larger expectation value in Eq. (2.25), making them important. However, they are rare events if directly sampled from Eq. (2.26). The incorporation of multi-level splitting facilitates a more efficient search for these pivotal permutations, resulting in reduced variance. By contrast, pMCS can be embedded into the multi-level splitting method proposed in Murray et al. [104] for estimating the conditional probabilities $\{\frac{\Pr(F_l)}{\Pr(F_{l-1})}\}_{l=2}^{M}$ [23].

2.3.5 Importance sampling

We consider the case where the goal is to estimate the probability that the input vector $\boldsymbol{X}$ falls in the failure domain F, denoted as $\Pr(\boldsymbol{X} \in F)$. In network reliability assessment, the domain F is characterized by the network performance function $g(\boldsymbol{X})$ and a specified threshold γ. The precise definition of F depends on the problem at hand and has been outlined in Section 2.1.

The basic idea of importance sampling (IS) is to sample from a proposal distribution, also known as IS distribution, under which the rare event is more likely to occur and to correct the resulting bias in the estimate by multiplying each sample in the IS estimator with an appropriate likelihood ratio L [82]. Let $p_{IS}(\boldsymbol{x})$ be the IS distribution and $\{\boldsymbol{x}^{(k)}\}_{k=1}^{N}$ be the N samples generated from $p_{IS}(\boldsymbol{x})$. The IS estimator of $\Pr(\boldsymbol{X} \in F)$ can then be written as

$$\widehat{p}_f^{(IS)} = \frac{1}{N}\sum_{k=1}^{N} \mathbb{I}\{\boldsymbol{x}^{(k)} \in F\} \frac{p_{\boldsymbol{X}}(\boldsymbol{x}^{(k)})}{p_{IS}(\boldsymbol{x}^{(k)})}, \tag{2.36}$$

where the likelihood ratio (or IS weight) $L(\boldsymbol{x}) \triangleq \frac{p_{\boldsymbol{X}}(\boldsymbol{x})}{p_{IS}(\boldsymbol{x})}$ can be interpreted as an adjustment factor that compensates for the fact that samples are generated from $p_{IS}(\boldsymbol{x})$ instead of $p_{\boldsymbol{X}}(\boldsymbol{x})$ [110]. The IS estimator in Eq. (2.36) is unbiased if the intersection of the failure domain F and the support of $p_{\boldsymbol{X}}(\boldsymbol{x})$ is included in the support of $p_{IS}(\boldsymbol{x})$ [110]. The variance of the estimator crucially depends on the choice of the IS distribution. A proper choice of the IS distribution can lead to a significantly smaller variance than that of crude MCS. Indeed, the theoretical optimal IS distribution $p_{IS}^*(\boldsymbol{x})$ that results in zero variance of the estimator is equal to the input distribution conditional on F. That is

$$p_{IS}^*(\boldsymbol{x}) = \frac{p_{\boldsymbol{X}}(\boldsymbol{x})\mathbb{I}\{\boldsymbol{x} \in F\}}{p_f} \triangleq p_{\boldsymbol{X}}(\boldsymbol{x}|F). \tag{2.37}$$

Unfortunately, $p_{IS}^*(\boldsymbol{x})$ cannot be directly used since its analytical expression relies on a prior knowledge of the sought failure probability p_f. Nevertheless, the optimal IS distribution $p_{IS}^*(\boldsymbol{x})$ still provides guidance for selecting an appropriate IS distribution. A common approach is to perform an initial first/second-order reliability method analysis [100] or employ a Markov chain simulation algorithm [6] to form a distribution that resembles $p_{IS}^*(\boldsymbol{x})$. Alternatively, one can approximate $p_{IS}^*(\boldsymbol{x})$ in an adaptive manner through the application of the CE methods [122, 123].

In connectivity-based problems, the input vector $\boldsymbol{X} = (X_1, \cdots, X_n)$ is binary, indicating the states of network edges: 0 for failure and 1 for operational. The network fails when it is disconnected. For such cases, L'Ecuyer et al. [86] proved that the optimal IS distribution $p_{IS}^*(\boldsymbol{x})$ can be reformulated as follows:

$$\begin{aligned} \widetilde{p}_d &= p_d \frac{u(x_1, \cdots, x_{d-1}, x_d = 0)}{u(x_1, \cdots, x_{d-1})}, \quad d = 1 \cdots n, \\ p_{IS}^*(\boldsymbol{x}) &= \prod_{d=1}^{n} \widetilde{p}_d^{\mathbb{I}\{x_d=0\}} (1 - \widetilde{p}_d)^{\mathbb{I}\{x_d=1\}}, \end{aligned} \tag{2.38}$$

where $p_d = \Pr(X_d = 0)$ is the failure probability of the d-th edge, and $u(x_1, \cdots, x_d)$ denotes the network failure probability given $X_1 = x_1, \cdots, X_d = x_d$ ($u(\emptyset)$ denotes the unconditional failure probability.) Eq. (2.38) suggests a sequential sampling scheme to obtain samples from the optimal IS distribution: first sample X_1 from the Bernoulli distribution $\mathrm{Ber}(x; 1 - \widetilde{p}_1)$, then given

x_1 sample X_2 from $\mathrm{Ber}(x; 1-\widetilde{p}_2)$ and so forth. However, this approach is impractical as $u(\emptyset)$ and $\{u(x_1, \cdots, x_d)\}_{d=1}^{n-1}$ are unknown. To circumvent this issue, the approximate zero-variance importance sampling (AZIS) method replaced $u(x_1, \cdots, x_d)$ by its approximation $\widehat{u}(x_1, \cdots, x_d)$ [86]. Specifically, one identifies the minimal cut that has the highest probability in $\mathcal{G}_d$, a graph formed by removing failed edges and merging functional edges in $x_1, \cdots, x_d$, and takes its probability as $\widehat{u}(x_1, \cdots, x_d)$. This minimal cut can be identified in polynomial time after equipping each i-th edge in $\mathcal{G}_d$ with the weight $-\ln(p_i)$ [127]. More importantly, under this minimal-cut-maximum-probability approximation strategy, the coefficient of variation of the resulting AZIS estimator is proven to be bounded. The AZIS method can be enhanced by integrating the graph reduction techniques, such as the series-parallel reduction [87]. Moreover, it can be extended to account for node failures, which are commonly encountered when analyzing railway telecommunication systems [118].

The multi-level CE method [122, 82] provides an alternative approach for approximating the optimal IS distribution $p^*_{IS}(\boldsymbol{x})$. Its fundamental idea involves iterative minimization of the Kullback-Leibler (KL) divergence between a parametric model and a sequence of intermediate target distributions that gradually approach $p_{\boldsymbol{X}}(\boldsymbol{x})$. The distribution obtained in the final iteration is employed as the IS distribution. For network reliability problems with binary or multi-state input, the obvious choice of the parametric model is the multivariate categorical distribution. However, as will be illustrated later in Chapter 5, there is a significant overfitting issue when the edge failure probability is small. Also, as the network performance is discrete, the original adaptation of the intermediate target distributions is not always feasible and can even get stuck. In the literature, the two issues are avoided by integrating the creation process into the multi-level CE method [71]. Consequently, the discrete network performance is transformed into a continuous one, the critical time T^*, with input now becoming a set of exponentially distributed repair times. The product of independent exponential distributions can then be employed as the parametric model [71]. Since the IS distribution of the repair time obtained by the multi-level CE method also defines an IS distribution for the permutations in pMCS, it is therefore natural to consider combining both methods. Note that there are network reliability problems that have a continuous performance. e.g., the shortest path problem, for which the CE method is directly applicable [66].

2.4 Other widely used approaches

In this section, we discuss other methods for network reliability assessment.

2.4.1 Probability density evolution method

The probability density evolution method (PDEM) was initially developed for the analysis of nonlinear stochastic dynamic systems [90]. This method captures the distribution of the structural response and accommodates the uncertainty inherent in both structural properties and external excitations. At the core of this method lies the generalized density evolution equation (or Li-Chen equation). For specific systems, solving this equation concurrently with the structural dynamic equations, the closed-form solution of the response's probability density function (PDF) can be derived [78]. For general systems, however, only numerical solutions are accessible. Specifically, the numerical method begins with a set of representative points selected to minimize a specified discrepancy, such as the

GF discrepancy [35]. These representative points partition the sample space of the random parameters into disjoint subdomains, each associated with a representative point. Within each subdomain, the time derivative of the response (e.g., velocity is the time derivative of the displacement) is first calculated through deterministic dynamic analysis, with the random parameters in dynamic structural functions fixed at the associated representative point. The resulting derivative trajectories are then used for driving the finite difference methods for numerically solving the generalized density evolution equation. This is called the point evolution scheme, and the efficiency hinges on the selection of the representative points set and the configuration of the finite difference scheme such as the grid size [132]. If the performance of infrastructure networks is continuous and governed by stochastic dynamic equations, the PDEM can be employed for assessing the functional reliability of these networks. This idea has already been explored for buried pipeline systems [98, 97] and water supply systems [102, 101].

2.4.2 Matrix-based system reliability method

The matrix-based system reliability method is another practically efficient method [80] for precisely calculating the failure probability of networks. This method necessitates an explicit expression of the failure event F as the union, intersections, or complements of the component events $E_{i,j}$, which denotes the i-th component in the j-th state. The failure event vector $\boldsymbol{c}^F$ can then be efficiently constructed through matrix operation [39] of component event vectors $\boldsymbol{c}^{E_{i,j}}$, which are often easy to construct. Even for multi-state components, these event vectors are binary, indicating whether or not each system state belongs to F or $E_{i,j}$ [80]. For instance, in a system consisting of n tri-state components, the total number of system states amounts to 3^n, so $\boldsymbol{c}^F$ (resp. $\boldsymbol{c}^{E_{i,j}}$) is a binary vector of size 3^n, and if the k-th system state belongs to F (resp. $E_{i,j}$), $c_k^F = 1$ (resp. $c_k^{E_{i,j}} = 1$). For binary components, however, the matrix operation for computing $\boldsymbol{c}^F$ can be substituted with more efficient logical operations [39]. Let the vector $\boldsymbol{p}^{\Omega_{\boldsymbol{X}}}$ collect the probability of each system state in $\Omega_{\boldsymbol{X}}$, which can be easily calculated when network components are independent or subject to common cause failure [80]. Finally, the failure probability is calculated by $(\boldsymbol{c}^F)^{\mathbf{T}} \cdot \boldsymbol{p}^{\Omega_{\boldsymbol{X}}}$.

2.4.3 Linear programming and multi-scale decomposition

In cases where $\boldsymbol{c}^F$ can be efficiently constructed, yet the description of the input distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$ is incomplete, obtaining the narrowest bounds of the failure probability involves solving a linear programming problem [130]. This can happen, for instance, when the dependence structure of network components is unknown, and only the marginal (or uni-component in Kang et al. [80]) distributions are available. However, the number of optimization variables grows exponentially in the number of network components, which restricts the applicability of this bounding technique to networks of limited size. Through incorporating a multi-scale decomposition of the network [45], the linear programming technique can be applied to larger networks. The basic idea of the multi-scale decomposition is to introduce a set of super-components, each comprising one or a few network components, and reformulate the failure event as an intersection, union, or complement of super-component events, which in turn should be represented as a function of component events. The linear programming problem is then addressed to first determine the probability bounds of super-component events; these established bounds subsequently serve as constraints for bounding the final

failure probability. In this way, the initial linear programming problem, characterized by a large number of optimization variables, is transformed into several smaller nested linear programming problems, thereby significantly reducing the computational cost. As a trade-off, the resulting bound is no longer the tightest. Note that the selection of super-components significantly influences the overall efficiency of the method, and an inappropriate choice can even lead to less efficient results compared to the crude approach. Also, besides linear programming, the multi-scale modeling of the network can also be incorporated with other network reliability methods [94, 89, 131].

2.4.4 System and survival signatures

When inputs are binary and exchangeable, i.e., they follow an independently and identically distributed Bernoulli distribution, network reliability can be characterized by using the system signature [124]. The signature is defined as the portion of the functional system states among those with a specified number of functional components. In particular, let us consider a n-component system with component failure probability denoted as p. Assume that all system states possessing i functional components and resulting in the system functionality are included in the set S_i. The system signature $\Phi(i)$ can then be expressed as $\frac{|S_i|}{\binom{n}{i}}$ for each $i = 0, \cdots, n$, where the denominator $\binom{n}{i}$ is the binomial coefficient computing the total number of states with exact i functional components, and $|S_i|$ is the cardinality of S_i. Due to the exchangeability of the components, the probability of these states are all equal, so the signature $\Phi(i)$ can be interpreted as the conditional reliability given exactly i functional components. Consequently, according to the total probability theorem, the reliability can be calculated as

$$1 - p_f = \sum_{i=1}^{n} \Phi(i) \binom{n}{i} p^{n-i} (1-p)^i. \tag{2.39}$$

Note that $\Phi(i)$ is decoupled from the input distribution, making Eq. (2.39) particularly suitable for time-dependent reliability assessment, e.g., for computing the survival function of the system. In this context, the component failure probability is often characterized by a random time to failure, but the network performance function remains the same, so $\Phi(i)$ only needs to be calculated once. When there are $K > 1$ types of components, and the exchangeability is confined to components of the same type, the system signature is generalized to the survival signature $\Phi(i_1, \cdots, i_K)$ [40], which is defined as the fraction of the functional system states conditional on i_1 functional components of type 1, i_2 functional components of type 2, and so forth. Similarly to the system signature, $\Phi(i_1, \cdots, i_K)$ can also be interpreted as conditional reliability and is also decoupled from the probabilistic inputs. Estimating the survival signature can be accomplished through analytical methods, such as ROBDD [121], or by utilizing sampling-based algorithms [14, 46]. For large-scale networks with multiple types of components, the number of signatures is typically large, and an efficient estimation of these signatures still remains an open question. Survival signatures can also be expanded to address issues involving multi-state inputs [117] or dependent component failures [61].

In static network reliability settings, the estimation of system signatures through sampling can be integrated into a broader stratified sampling framework, where each stratum is characterized by a specified number of failed components. Van Slyke and Frank [134] have previously explored this idea for independent and identical inputs and utilized a proportional allocation strategy for distributing the computation budget. As for independent but non-identical inputs, the stratified sampling approach is still in development.

2.5 Conclusions and summary

We have reviewed the state-of-the-art network reliability methods. To conclude, a concise overview of these methods is provided in Table 2.3. The table is self-explanatory, constructed from the information provided in this chapter.

Table 2.3: A brief overview of network reliability methods detailed in Chapter 2

	subsections	dependent inputs	multi-state inputs	coherent g	error metric	generally unsuitable for
BDD	2.2.1	adaptable[1]	adaptable	not required	exact	large n, complex[4] g
cut(path)-based	2.2.2	adaptable	applicable[2]	required	exact(or bounds)	large n, complex[5] g
SSD	2.2.3	adaptable	applicable	required	reliability bounds	large n, high p_i
matrix-based	2.4.2	applicable	applicable	not required	exact	large n, complex[6] g
MCS	2.3.1	applicable	applicable	not required	MSE	small p_f
counting-based	2.3.2	unknown[3]	unknown	not required	(ϵ, δ)-approx.	small p_i, complex[7] g
pMCS	2.3.3	unknown	applicable	required	MSE	costly g, small p_f
creation process +splitting	2.3.4.4	unknown	applicable	required	MSE	costly g
creation process +CE	2.3.5	unknown	applicable	required	MSE	costly g
SuS	2.3.4.4	applicable	applicable	not required	MSE	g with high discontinuity
CE	2.3.5	applicable	applicable	not required	MSE	small p_i
signatures	2.4.4	adaptable	adaptable	not required	exact(or MSE)	large n
PDEM	2.4.1	applicable	applicable	not required	error bounds	sensitive to the grid size

[1] The method needs to be adapted, and the required adaptation can be found in literature.
[2] The method can be applied directly, or if adaptation is necessary, the specific adjustments are detailed in Chapter 2.
[3] The method needs to be adapted, and the required adaptation is not known by the author.
[4] The performance metric g without an efficient construction of its ROBDD.
[5] The performance metric g without an efficient searching algorithm for all maximum lower vectors (or minimal upper vectors).
[6] The performance metric g, for which the failure event F cannot be explicitly expressed as the union, intersection, or complements of the component events $E_{i,j}$.
[7] The performance metric g that cannot be encoded as a Σ_1^1 form.

In the following, we compare different network reliability methods and also highlight their respective limitations when estimating rare events in infrastructure networks. In particular, calculating the failure probability of general networks is NP-hard, and no efficient exact algorithms are believed to exist. Although for connectivity-based performance metrics, practically efficient approaches, such as ROBDD, cut(path)-based, and matrix-based methods can be adopted, their adaptation to more complex metrics is non-trivial. Only a few network performance metrics enable efficient construction of the ROBDD or an efficient searching algorithm for all maximum lower vectors (or minimal upper vectors), and in most cases, the failure event cannot be expressed as the union, intersection, or complement of the component events. While deterministic approximation approaches, such as SSD, are significantly more general and flexible, their efficiency is rooted in the coherency or monotonicity of the performance metric, whereby one calculation of the metric is sufficient to rule out a number of states. Physics-driven metrics, however, do not necessarily admit such a monotonic structure, and the justification can be tricky. Moreover, the methods mentioned above are not scalable in high dimensions except under specific conditions, e.g., when a well-structured ROBDD can be efficiently constructed, or when the probability mass is concentrated in a few states. On the other hand, they provide reliable error estimates, either exact value or deterministic bounds of p_f.

By contrast, the stochastic approximation, or sampling-based methods are more efficient in high dimensions and are natural choices for handling complex performance metrics. However, it may require a high number of samples to achieve an acceptable accuracy. Since critical infrastructures are designed to be highly reliable with a notably small failure probability, such a limitation is prominent, especially when calculating the performance metric is computationally intensive. Hence, variance reduction techniques become essential. The standard SuS may lead to significant error if the CDF of $g(\boldsymbol{X})$ contains notable discontinuities or 'jumps', while CE-based IS encounters the overfitting issue, particularly when the component failure probability p_i is small. To address these challenges, variance reduction techniques are frequently coupled with smoothing techniques, which transform the discrete metric into a continuous one in network reliability assessment. Some of these transformations are problem-specific and only apply to connectivity-based problems; others, such as the creation process, require multiple calculations of the original network performance metric, thus introducing significant additional costs. Consequently, creation-process-based methods, such as pMCS, creation process + CE, creation process + splitting are sensitive to costly network performance function g. The error associated with these sampling-based methods is often assessed through an estimated MSE of the failure probability estimator $\widehat{p}_f$, which can be inaccurate when $\widehat{p}_f$ is highly skewed. The counting-based method offers a more rigorous $(\epsilon - \delta)$-approximation of p_f, but it necessitates encoding the network performance metric g in a Σ_1^1 form, which is challenging for physics-based metrics. Also, the method will introduce significant computational costs if component failure probability p_i is small.

These limitations further justify the necessity of this thesis, which aims to bridge the gap between current network reliability methods and the challenges encountered in the reliability calculation of infrastructure networks. To this end, we extend two standard variance reduction techniques, multi-level splitting, and CE-based importance sampling, to handle the network reliability problems, and this will be elaborated in Part II of this thesis.

Chapter 3

Concluding remarks

3.1 Summary

In this thesis, we expand upon two widely adopted variance reduction techniques, specifically, SuS and CE-based importance sampling, to assess network reliability and enhance their performance in the context of physics-driven network performance metrics.

In Chapter 4, we propose an adaptive effort subset simulation (aE-SuS) algorithm designed to effectively address any significant discontinuities that may exist in network performance. When such discontinuity appears, standard SuS can lead to significant errors and even fail to reach the failure domain. To circumvent this issue, the aE-SuS algorithm adjusts the conditional probabilities based on the empirical conditional CDF at each level and adapts the number of samples accordingly to ensure an approximately equal number of seeds at each level. For binary network performance metrics, such as connectivity, the aE-SuS reduces to crude MCS. By contrast, if the network performance is almost continuous, aE-SuS behaves similarly to standard SuS. The efficiency of the aE-SuS algorithm for network reliability assessment relies on an MCMC algorithm that enables efficient sampling in discrete space. To this end, we present three different MCMC algorithms: the adaptive conditional sampler equipped with the Rosenblatt transformation, a novel independent Metropolis-Hasting algorithm (both introduced in Chapter 4), and the Gibbs sampler [146].

In Chapter 5, we identify the overfitting issue when fitting a categorical distribution in CE methods. To mitigate this issue, we introduce a consistent Bayesian estimator of the parameters of the categorical distribution, which incorporates prior information. This Bayesian estimator can be integrated into the iCE method for rare event estimation, resulting in an unbiased importance sampling distribution. We term this method as BiCE. In contrast to the proposed approach, the standard iCE method, despite its higher computational cost, can exhibit significant bias. The adaptation strategy of the iCE method is also substantiated by two theorems, which guarantee the convergence of the algorithm for network reliability assessment in ideal settings.

Since the independent categorical distribution cannot capture the dependence among network components, in Chapter 6, we update the parameters of a more flexible parametric distribution, specifically, the categorical mixture, so as to form a better approximation of the optimal importance

sampling distribution. The major contribution therein is a generalized version of the expectation-maximization algorithm to approximate the weighted MAP. The number of mixture components is determined as a by-product of the generalized EM algorithm using the Bayesian information criterion.

In Chapter 7, we provide unified benchmarks to streamline the comparison of different network reliability algorithms and to explore potential synergies between them. Specifically, we introduce a hybrid approach that leverages the respective strengths of both aE-SuS and annealed particle integration methods. The new approach is unbiased and works well for discrete network performance. Furthermore, we expand existing sensitivity methods to identify critical network components employing failure samples. For aE-SuS, we generalize the approach in Zwirglmaier et al. [146] and express the component importance measure as a function of the threshold. This is illustrated in Chapter 7. Similar ideas can be adapted to BiCE, as detailed in Chapter 6.

3.2 Outlook

Both multi-level splitting and CE-based importance sampling rely on sampling from a sequence of intermediate distribution that gradually approaches the failure domain. Ideally, we would be able to sample independently from the intermediate target distribution. In practice, however, MCMC sampling generates dependent samples, and the CE-based methods sample from a distribution that only resembles the target one. To enhance the efficiency of SuS or multi-level splitting methods, it is essential to develop MCMC algorithms that effectively sample in combinatorial spaces, and for CE-based methods, the key challenge lies in identifying a suitable approximation for the intermediate target distribution. The BiCE method provides a more flexible approach for updating distribution models by introducing a prior term or a penalization term in the CE-based optimization problem. However, the selection of an appropriate prior remains an open issue, necessitating further detailed investigation. Note that, although the methodologies developed in this thesis are generally applicable to a wide range of black-box systems, the underlying physics governing the network's performance is well understood. Thus, a logical step for future investigation involves leveraging embedded physics to devise more efficient MCMC algorithms and to identify good proxies of the intermediate target distributions. This demands collaboration with electrical engineering professionals and researchers. If the probabilistic input model possesses any intrinsic structure, it is also beneficial to leverage such information. As an example, exploring network reliability methods that account for common cause failures presents a promising direction.

Another promising avenue worth exploring is the development of network reliability algorithms that ensure a reliable error estimate. In practical applications, there is no crude MCS reference, and the true failure probability is also unknown. Thus, it is inefficient to merely produce an estimate of the failure probability; the quality of the estimate becomes equally essential. For adaptMCS methods, the quality is reflected by the estimated variance through one single run of the algorithm, but this variance estimate can be poor. Alternatively, these algorithms can be validated through benchmark models, but the extrapolation from the benchmarks to the real applications can be misleading. This further underlines the need for error-accountable adaptMCS algorithms, which ensures a reliable error estimate, e.g., an (ϵ, δ)-approximation of p_f.

As for other network reliability algorithms, future research can be conducted to address their limitations as illustrated in Table 2.3. This includes generalizing the creation-process-based or counting-based methods for dependent inputs, formulating efficient construction strategies in BDD for general metrics, developing smart searching algorithms for maximum lower vectors or minimal upper vectors, and exploring alternative smoothing techniques beyond creation process. Additionally, validating the coherency of specific physics-based metrics or transforming non-coherent metrics into coherent ones would also broaden the scope and applicability of methods grounded on coherency. In static network reliability assessment, system signatures can be further included into the stratified sampling framework. While the stratified sampling method is well-established for identical components, its extension to encompass non-identical or even dependent components seems unexplored, providing another entrance point for the future research.

Besides network failure probability, the criticality of network components also provides valuable insights for decision-making, reliability management, and resilience planning. In practice, the criticality are quantified by various component importance measures. Incorporating the calculation of these metrics into the network reliability methods outlined in Table 2.3 is therefore of great importance and worth exploring in the future.

References

[1] K. Aggarwal, K. Misra, and J. Gupta. "A fast algorithm for reliability evaluation". In: *IEEE Transactions on Reliability* 24.1 (1975), pp. 83–85.

[2] A. Agrawal and R. E. Barlow. "A survey of network reliability and domination theory". In: *Operations Research* 32.3 (1984), pp. 478–492.

[3] S. B. Akers. "Binary decision diagrams". In: *IEEE Transactions on Computers* 100.6 (1978), pp. 509–516.

[4] C. Alexopoulos. "A note on state-space decomposition methods for analyzing stochastic flow networks". In: *IEEE Transactions on Reliability* 44.2 (1995), pp. 354–357.

[5] C. Alexopoulos and G. S. Fishman. "Characterizing stochastic flow networks using the Monte Carlo method". In: *Networks* 21.7 (1991), pp. 775–798.

[6] S.-K. Au and J. L. Beck. "A new adaptive importance sampling scheme for reliability calculations". In: *Structural Safety* 21.2 (1999), pp. 135–158.

[7] S.-K. Au and J. L. Beck. "Estimation of small failure probabilities in high dimensions by subset simulation". In: *Probabilistic Engineering Mechanics* 16.4 (2001), pp. 263–277.

[8] S.-K. Au and Y. Wang. *Engineering Risk Assessment with Subset Simulation*. John Wiley & Sons, 2014.

[9] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. "Algebric decision diagrams and their applications". In: *Formal methods in system design* 10 (1997), pp. 171–206.

[10] G. Bai, M. J. Zuo, and Z. Tian. "Ordering heuristics for reliability evaluation of multistate networks". In: *IEEE Transactions on Reliability* 64.3 (2015), pp. 1015–1023.

[11] M. O. Ball. "Computational complexity of network reliability analysis: An overview". In: *IEEE Transactions on Reliability* 35.3 (1986), pp. 230–239.

[12] M. O. Ball, C. J. Colbourn, and J. S. Provan. "Network reliability". In: *Handbooks in Operations Research and Management Science* 7 (1995), pp. 673–762.

[13] M. O. Ball and G. L. Nemhauser. "Matroids and a reliability analysis problem". In: *Mathematics of Operations Research* 4.2 (1979), pp. 132–143.

[14] J. Behrensdorf, T.-E. Regenhardt, M. Broggi, and M. Beer. "Numerically efficient computation of the survival signature for the reliability analysis of large networks". In: *Reliability Engineering & System Safety* 216 (2021), p. 107935.

[15] Z. I. Botev and D. P. Kroese. "An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting". In: *Methodology and Computing in Applied Probability* 10.4 (2008), pp. 471–505.

[16] Z. I. Botev and D. P. Kroese. "Efficient Monte Carlo simulation via the generalized splitting method". In: *Statistics and Computing* 22.1 (2012), pp. 1–16.

[17] Z. I. Botev, P. L'Ecuyer, G. Rubino, R. Simard, and B. Tuffin. "Static network reliability estimation via generalized splitting". In: *INFORMS Journal on Computing* 25.1 (2013), pp. 56–71.

[18] Z. I. Botev, P. L'Ecuyer, and B. Tuffin. "Reliability estimation for networks with minimal flow demand and random link capacities". In: *arXiv preprint arXiv:1805.03326* (2018).

[19] C.-E. Bréhier and T. Lelièvre. "On a new class of score functions to estimate tail probabilities of some stochastic processes with adaptive multilevel splitting". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.3 (2019), p. 033126.

[20] J. I. Brown, C. J. Colbourn, D. Cox, C. Graves, and L. Mol. "Network reliability: Heading out on the highway". In: *Networks* 77.1 (2021), pp. 146–160.

[21] R. E. Bryant. "Graph-based algorithms for Boolean function manipulation". In: *IEEE Transactions on Computers* 100.8 (1986), pp. 677–691.

[22] J. W. Busby, K. Baker, M. D. Bazilian, A. Q. Gilbert, E. Grubert, V. Rai, J. D. Rhodes, S. Shidore, C. A. Smith, and M. E. Webber. "Cascading risks: Understanding the 2021 winter blackout in Texas". In: *Energy Research & Social Science* 77 (2021), p. 102106.

[23] H. Cancela, L. Murray, F. Robledo, P. Romero, and P. Sartor. "On the reliability estimation of stochastic binary systems". In: *International Transactions in Operational Research* 29.3 (2022), pp. 1688–1722.

[24] H. Cancela, L. Murray, and G. Rubino. "Efficient estimation of stochastic flow network reliability". In: *IEEE Transactions on Reliability* 68.3 (2019), pp. 954–970.

[25] H. Cancela, L. Murray, and G. Rubino. "Reliability estimation for stochastic flow networks with dependent arcs". In: *IEEE Transactions on Reliability* (2022), pp. 622–636.

[26] F. Cérou, P. Del Moral, T. Furon, and A. Guyader. "Sequential Monte Carlo for rare event estimation". In: *Statistics and Computing* 22.3 (2012), pp. 795–808.

[27] F. Cérou and A. Guyader. "Adaptive multilevel splitting for rare event analysis". In: *Stochastic Analysis and Applications* 25.2 (2007), pp. 417–443.

[28] F. Cérou, A. Guyader, T. Lelievre, and D. Pommier. "A multiple replica approach to simulate reactive trajectories". In: *The Journal of Chemical Physics* 134.5 (2011), p. 054108.

[29] F. Cérou, A. Guyader, and M. Rousset. "Adaptive multilevel splitting: Historical perspective and recent results". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.4 (2019), p. 043108.

[30] J. Chan, I. Papaioannou, and D. Straub. "An adaptive subset simulation algorithm for system reliability analysis with discontinuous limit states". In: *Reliability Engineering & System Safety* 225 (2022), p. 108607.

[31] J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method for network reliability assessment". In: *Structural Safety* 103 (2023), p. 102344.

[32] J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method with categorical mixture models". (Under review).

[33] J. Chan, R. Paredes, I. Papaioannou, L. Duenas-Osorio, and D. Straub. "Adaptive Monte Carlo methods for estimating rare events in power grids". (Under review).

[34] S. K. Chaturvedi. *Network reliability: Measures and evaluation*. John Wiley & Sons, 2016.

[35] J. Chen and S. Zhang. "Improving point selection in cubature by a new discrepancy". In: *SIAM Journal on Scientific Computing* 35.5 (2013), A2121–A2149.

[36] H. Chernoff. "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations". In: *The Annals of Mathematical Statistics* (1952), pp. 493–507.

[37] J. Ching, S.-K. Au, and J. L. Beck. "Reliability estimation for dynamical systems subject to stochastic excitation using subset simulation with splitting". In: *Computer Methods in Applied Mechanics and Engineering* 194.12-16 (2005), pp. 1557–1579.

[38] J. Ching and W.-C. Hsu. "An efficient method for evaluating origin-destination connectivity reliability of real-world lifeline networks". In: *Computer-Aided Civil and Infrastructure Engineering* 22.8 (2007), pp. 584–596.

[39] J. Contreras-Jiménez, F. Rivas-Dávalos, J. Song, and J. Guardado. "Multi-state system reliability analysis of HVDC transmission systems using matrix-based system reliability method". In: *International Journal of Electrical Power & Energy Systems* 100 (2018), pp. 265–278.

[40] F. P. Coolen and T. Coolen-Maturi. "Generalizing the signature to systems with multiple types of components". In: *Complex Systems and Dependability*. Springer. 2012, pp. 115–130.

[41] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White. "MCMC methods for functions: Modifying old algorithms to make them faster". In: *Statistical Science* 28.3 (2013), pp. 424–446.

[42] M. S. Daly and C. Alexopoulos. "State-space partition techniques for multiterminal flows in stochastic networks". In: *Networks: An International Journal* 48.2 (2006), pp. 90–111.

[43] P. Del Moral. *Feynman-Kac Formulae*. Springer, 2004.

[44] N. Deo and M. Medidi. "Parallel algorithms for terminal-pair reliability". In: *IEEE Transactions on Reliability* 41.2 (1992), pp. 201–209.

[45] A. Der Kiureghian and J. Song. "Multi-scale reliability analysis and updating of complex systems by use of linear programming". In: *Reliability Engineering & System Safety* 93.2 (2008), pp. 288–297.

[46] F. Di Maio, C. Pettorossi, and E. Zio. "Entropy-driven Monte Carlo simulation method for approximating the survival signature of complex infrastructures". In: *Reliability Engineering & System Safety* 231 (2023), p. 108982.

[47] W. Dotson and J. Gobien. "A new analysis technique for probabilistic graphs". In: *IEEE Transactions on Circuits and Systems* 26.10 (1979), pp. 855–865.

[48] P. Doulliez and E. Jamoulle. "Transportation networks with random arc capacities". In: *Revue Française d'Automatique, Informatique, Recherche Opérationnelle. Recherche Opérationnelle* 6.V3 (1972), pp. 45–59.

[49] L. Duenas-Osorio, K. Meel, R. Paredes, and M. Vardi. "Counting-based reliability estimation for power-transmission grids". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[50] M. C. Easton and C. Wong. "Sequential destruction method for Monte Carlo evaluation of system reliability". In: *IEEE Transactions on Reliability* 29.1 (1980), pp. 27–32.

[51] *Electric Disturbance Events (OE-417) Annual Summaries.*

[52] T. Elperin, I. Gertsbakh, and M. Lomonosov. "An evolution model for Monte Carlo estimation of equilibrium network renewal parameters". In: *Probability in the Engineering and Informational Sciences* 6.4 (1992), pp. 457–469.

[53] T. Elperin, I. Gertsbakh, and M. Lomonosov. "Estimation of network reliability using graph evolution models". In: *IEEE Transactions on Reliability* 40.5 (1991), pp. 572–581.

[54] G. S. Fishman. "A Monte Carlo sampling plan for estimating network reliability". In: *Operations Research* 34.4 (1986), pp. 581–594.

[55] G. S. Fishman and T.-Y. D. Shaw. "Evaluating reliability of stochastic flow networks". In: *Probability in the Engineering and Informational Sciences* 3.4 (1989), pp. 493–509.

[56] L. Fratta and U. Montanari. "A Boolean algebra method for computing the terminal reliability in a communication network". In: *IEEE Transactions on Circuit Theory* 20.3 (1973), pp. 203–211.

[57] M. J. Garvels and D. P. Kroese. "A comparison of RESTART implementations". In: *1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*. Vol. 1. IEEE. 1998, pp. 601–608.

[58] M. J. Garvels, J.-K. C. Van Ommeren, and D. P. Kroese. "On the importance function in splitting simulation". In: *European Transactions on Telecommunications* 13.4 (2002), pp. 363–371.

[59] M. J. J. Garvels. "The splitting method in rare event simulation". PhD thesis. University of Twente, 2000.

[60] A. E. Gelfand and A. F. Smith. "Sampling-based approaches to calculating marginal densities". In: *Journal of the American statistical association* 85.410 (1990), pp. 398–409.

[61] H. George-Williams, G. Feng, F. P. Coolen, M. Beer, and E. Patelli. "Extending the survival signature paradigm to complex systems with non-repairable dependent failures". In: *Proceedings of the Institution of Mechanical Engineers, part O: journal of risk and reliability* 233.4 (2019), pp. 505–519.

[62] I. Gertsbakh, E. Neuman, and R. Vaisman. "Monte Carlo for estimating exponential convolution". In: *Communications in Statistics-Simulation and Computation* 44.10 (2015), pp. 2696–2704.

[63] I. Gertsbakh, R. Rubinstein, Y. Shpungin, and R. Vaisman. "Permutational methods for performance analysis of stochastic flow networks". In: *Probability in the Engineering and Informational Sciences* 28.1 (2014), pp. 21–38.

[64] I. B. Gertsbakh and Y. Shpungin. *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*. CRC press, 2016.

[65] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. "Multilevel splitting for estimating rare event probabilities". In: *Operations Research* 47.4 (1999), pp. 585–600.

[66] A. A. Gouda and T. Szántai. "Rare event probabilities in stochastic networks". In: *Central European Journal of Operations Research* 16.4 (2008), pp. 441–461.

[67] A. Guyader, N. Hengartner, and E. Matzner-Løber. "Simulation and estimation of extreme quantiles and extreme probabilities". In: *Applied Mathematics & Optimization* 64.2 (2011), pp. 171–196.

[68] G. Hardy, C. Lucet, and N. Limnios. "K-terminal network reliability measures with binary decision diagrams". In: *IEEE Transactions on Reliability* 56.3 (2007), pp. 506–515.

[69] J. He. "An extended recursive decomposition algorithm for dynamic seismic reliability evaluation of lifeline networks with dependent component failures". In: *Reliability Engineering & System Safety* 215 (2021), p. 107929.

[70] K.-P. Hui, N. Bean, M. Kraetzl, and D. Kroese. "Network reliability estimation using the tree cut and merge algorithm with importance sampling". In: *Proceedings of the 4th International Workshop on Design of Reliable Communication Networks*. IEEE. 2003, pp. 254–262.

[71] K.-P. Hui, N. Bean, M. Kraetzl, and D. P. Kroese. "The cross-entropy method for network reliability estimation". In: *Annals of Operations Research* 134.1 (2005), pp. 101–118.

[72] H. Imai, K. Sekine, and K. Imai. "Computational investigations of all-terminal network reliability via BDDs". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 82.5 (1999), pp. 714–721.

[73] J. A. Jacobson. "State space partitioning methods for solving a class of stochastic network problems". PhD thesis. Georgia Institute of Technology, 1993.

[74] X. Janan. "On multistate system analysis". In: *IEEE Transactions on Reliability* 34.4 (1985), pp. 329–337.

[75] C.-C. Jane, J.-S. Lin, and J. Yuan. "Reliability evaluation of a limited-flow network in terms of minimal cutsets". In: *IEEE Transactions on Reliability* 42.3 (1993), pp. 354–361.

[76] C.-C. Jane and Y.-W. Laih. "A practical algorithm for computing multi-state two-terminal reliability". In: *IEEE Transactions on Reliability* 57.2 (2008), pp. 295–302.

[77] H. A. Jensen and D. J. Jerez. "A stochastic framework for reliability and sensitivity analysis of large scale water distribution networks". In: *Reliability Engineering & System Safety* 176 (2018), pp. 80–92.

[78] Z. Jiang and J. Li. "Analytical solutions of the generalized probability density evolution equation of three classes stochastic systems". In: *Chinese Journal of Theoretical and Applied Mechanics* 48.2 (2016), pp. 413–421.

[79] H. Kahn and T. E. Harris. "Estimation of particle transmission by random sampling". In: *National Bureau of Standards Applied Mathematics Series* 12 (1951), pp. 27–30.

[80] W.-H. Kang, J. Song, and P. Gardoni. "Matrix-based system reliability method and applications to bridge networks". In: *Reliability Engineering & System Safety* 93.11 (2008), pp. 1584–1593.

[81] R. M. Karp and M. Luby. "Monte Carlo algorithms for the planar multiterminal network reliability problem". In: *Journal of Complexity* 1.1 (1985), pp. 45–64.

[82] D. P. Kroese, T. Taimre, and Z. I. Botev. *Handbook of Monte Carlo Methods*. Vol. 706. John Wiley & Sons, 2013.

[83] H. Kumamoto, K. Tanaka, and K. Inoue. "Efficient evaluation of system reliability by Monte Carlo method". In: *IEEE Transactions on Reliability* 26.5 (1977), pp. 311–315.

[84] H. Kumamoto, K. Tanaka, K. Inoue, and E. J. Henley. "Dagger-sampling Monte Carlo for system unavailability evaluation". In: *IEEE Transactions on Reliability* 29.2 (1980), pp. 122–125.

[85] P. L'Ecuyer, V. Demers, and B. Tuffin. "Rare events, splitting, and quasi-Monte Carlo". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 17.2 (2007), 9–es.

[86] P. L'Ecuyer, G. Rubino, S. Saggadi, and B. Tuffin. "Approximate zero-variance importance sampling for static network reliability estimation". In: *IEEE Transactions on Reliability* 60.3 (2011), pp. 590–604.

[87] P. L'Ecuyer, S. Saggadi, and B. Tuffin. "Graph reductions to speed up importance sampling-based static reliability estimation". In: *Proceedings of the 2011 Winter Simulation Conference (WSC)*. IEEE. 2011, pp. 429–438.

[88] C.-Y. Lee. "Representation of switching circuits by binary-decision programs". In: *The Bell System Technical Journal* 38.4 (1959), pp. 985–999.

[89] D. Lee and J. Song. "Multi-scale seismic reliability assessment of networks by centrality-based selective recursive decomposition algorithm". In: *Earthquake Engineering & Structural Dynamics* 50.8 (2021), pp. 2174–2194.

[90] J. Li and J. Chen. "Probability density evolution method for dynamic response analysis of structures with uncertain parameters". In: *Computational Mechanics* 34.5 (2004), pp. 400–409.

[91] J. Li and J. He. "A recursive decomposition algorithm for network seismic reliability evaluation". In: *Earthquake Engineering & Structural Dynamics* 31.8 (2002), pp. 1525–1539.

[92] W. Li. *Reliability Assessment of Electric Power Systems Using Monte Carlo Methods*. Springer Science & Business Media, 2013.

[93] H.-W. Lim and J. Song. "Efficient risk assessment of lifeline networks under spatially correlated ground motions using selective recursive decomposition algorithm". In: *Earthquake Engineering & Structural Dynamics* 41.13 (2012), pp. 1861–1882.

[94] H.-W. Lim, J. Song, and N. Kurtz. "Seismic reliability assessment of lifeline networks using clustering-based multi-scale approach". In: *Earthquake Engineering & Structural Dynamics* 44.3 (2015), pp. 355–369.

[95] J.-S. Lin, C.-C. Jane, and J. Yuan. "On reliability evaluation of a capacitated-flow network in terms of minimal pathsets". In: *Networks* 25.3 (1995), pp. 131–138.

[96] A. Lisnianski and G. Levitin. *Multi-state System Reliability: Assessment, Optimization and Applications*. World scientific, 2003.

[97] W. Liu, Z. Li, Z. Song, and J. Li. "Seismic reliability evaluation of gas supply networks based on the probability density evolution method". In: *Structural safety* 70 (2018), pp. 21–34.

[98] W. Liu, Q. Sun, H. Miao, and J. Li. "Nonlinear stochastic seismic analysis of buried pipeline systems". In: *Soil Dynamics and Earthquake Engineering* 74 (2015), pp. 69–78.

[99] M. O. Locks. "A minimizing algorithm for sum of disjoint products". In: *IEEE Transactions on Reliability* 36.4 (1987), pp. 445–453.

[100] H. O. Madsen, S. Krenk, and N. C. Lind. *Methods of Structural Safety*. Courier Corporation, 2006.

[101] H. Miao and J. Li. "Serviceability evaluation of water supply networks under seismic loads utilizing their operational physical mechanism". In: *Earthquake Engineering and Engineering Vibration* 21.1 (2022), pp. 283–296.

[102] H. Miao, W. Liu, and J. Li. "Seismic reliability analysis of water distribution networks on the basis of the probability density evolution method". In: *Structural Safety* 86 (2020), p. 101960.

[103] A. H. Al-Mohy and N. J. Higham. "A new scaling and squaring algorithm for the matrix exponential". In: *SIAM Journal on Matrix Analysis and Applications* 31.3 (2010), pp. 970–989.

[104] L. Murray, H. Cancela, and G. Rubino. "A splitting algorithm for network reliability estimation". In: *IIE Transactions* 45.2 (2013), pp. 177–189.

[105] J. Nahman. "Minimal paths and cuts of networks exposed to common-cause failures". In: *IEEE Transactions on Reliability* 41.1 (1992), pp. 76–80.

[106] R. M. Neal. "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011, pp. 113–163.

[107] L. D. Nel and C. J. Colbourn. "Combining Monte Carlo estimates and bounds for network reliability". In: *Networks* 20.3 (1990), pp. 277–298.

[108] M. Newman. *Networks*. Oxford University Press, 2018.

[109] Y.-F. Niu, Z.-Y. Gao, and W. H. Lam. "A new efficient algorithm for finding all d-minimal cuts in multi-state networks". In: *Reliability Engineering & System Safety* 166 (2017), pp. 151–163.

[110] A. B. Owen. *Monte Carlo Theory, Methods and Examples*. Standford, 2013.

[111] I. Papaioannou, W. Betz, K. Zwirglmaier, and D. Straub. "MCMC algorithms for subset simulation". In: *Probabilistic Engineering Mechanics* 41 (2015), pp. 89–103.

[112] R. Paredes, L. Dueñas-Osorio, and I. Hernandez-Fajardo. "Decomposition algorithms for system reliability estimation with applications to interdependent lifeline networks". In: *Earthquake Engineering & Structural Dynamics* 47.13 (2018), pp. 2581–2600.

[113] R. Paredes, L. Dueñas-Osorio, K. S. Meel, and M. Y. Vardi. "Principled network reliability approximation: A counting-based approach". In: *Reliability Engineering & System Safety* 191 (2019), p. 106472.

[114] R. Paredes, H. Talebiyan, and L. Dueñas-Osorio. "Path-dependent reliability and resiliency of critical infrastructure via particle integration methods". In: *Proceedings of the 13th International Conference on Structural Safety & Reliability*. IASSAR. 2022.

[115] J. S. Provan and M. O. Ball. "Computing network reliability in time polynomial in the number of cuts". In: *Operations Research* 32.3 (1984), pp. 516–526.

[116] J. S. Provan and M. O. Ball. "The complexity of counting cuts and of computing the probability that a graph is connected". In: *SIAM Journal on Computing* 12.4 (1983), pp. 777–788.

[117] J. Qin and F. P. Coolen. "Survival signature for reliability evaluation of a multi-state system with multi-state components". In: *Reliability Engineering & System Safety* 218 (2022), p. 108129.

[118] A. Rai, R. C. Valenzuela, B. Tuffin, G. Rubino, and P. Dersin. "Approximate zero-variance importance sampling for static network reliability estimation with node failures and application to rail systems". In: *2016 Winter Simulation Conference (WSC)*. IEEE. 2016, pp. 3201–3212.

[119] M. Rausand and A. Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications*. John Wiley & Sons, 2003.

[120] A. Rauzy. "New algorithms for fault trees analysis". In: *Reliability Engineering & System Safety* 40.3 (1993), pp. 203–211.

[121] S. Reed. "An efficient algorithm for exact computation of system and survival signatures using binary decision diagrams". In: *Reliability Engineering & System Safety* 165 (2017), pp. 257–267.

[122] R. Y. Rubinstein. "Optimization of computer simulation models with rare events". In: *European Journal of Operational Research* 99.1 (1997), pp. 89–112.

[123] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, 2016.

[124] F. J. Samaniego. *System Signatures and Their Applications in Engineering Reliability*. Springer Science & Business Media, 2007.

[125] A. Satyanarayana and M. K. Chang. "Network reliability and the factoring theorem". In: *Networks* 13.1 (1983), pp. 107–120.

[126] A. Satyanarayana and A. Prabhakar. "New topological formula and rapid algorithm for reliability analysis of complex networks". In: *IEEE Transactions on Reliability* 27.2 (1978), pp. 82–100.

[127] R. Sedgewick. *Algorithms in C, Part 5: Graph Algorithms*. Pearson Education, 2001.

[128] A. Shrestha and L. Xing. "A logarithmic binary decision diagram-based method for multistate system analysis". In: *IEEE Transactions on Reliability* 57.4 (2008), pp. 595–606.

[129] J. Skilling. "Nested sampling for general Bayesian computation". In: *Bayesian Analysis* 1.4 (2006), pp. 833–860.

[130] J. Song and A. Der Kiureghian. "Bounds on system reliability by linear programming". In: *Journal of Engineering Mechanics* 129.6 (2003), pp. 627–636.

[131] J. Song and S.-Y. Ok. "Multi-scale system reliability analysis of lifeline networks under earthquake hazards". In: *Earthquake Engineering & Structural Dynamics* 39.3 (2010), pp. 259–279.

[132] W. Tao and J. Li. "An ensemble evolution numerical method for solving generalized density evolution equation". In: *Probabilistic Engineering Mechanics* 48 (2017), pp. 1–11.

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

[134] R. Van Slyke and H. Frank. "Network reliability analysis: Part 1". In: *Networks* 1.3 (1971), pp. 279–290.

[135] M. Villén-Altamirano, J. Villén-Altamirano, et al. "RESTART: A method for accelerating rare event simulations". In: *Queueing, Performance and Control in ATM (ITC-13)* (1991), pp. 71–76.

[136] J. Wilson. "An improved minimizing algorithm for sum of disjoint products (reliability theory)". In: *IEEE Transactions on Reliability* 39.1 (1990), pp. 42–45.

[137] L. Xing and S. V. Amari. *Binary Decision Diagrams and Extensions for System Reliability Analysis*. John Wiley & Sons, 2015.

[138] L. Xing and Y. Dai. "A new decision-diagram-based method for efficient analysis on multistate systems". In: *IEEE Transactions on Dependable and Secure Computing* 6.3 (2008), pp. 161–174.

[139] Y. Yoo and N. Deo. "A comparison of algorithms for terminal-pair reliability". In: *IEEE Transactions on Reliability* 37.2 (1988), pp. 210–215.

[140] X. Zang, D. Wang, H. Sun, and K. S. Trivedi. "A BDD-based algorithm for analysis of multistate systems with multistate components". In: *IEEE Transactions on Computers* 52.12 (2003), pp. 1608–1618.

[141] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education". In: *IEEE Transactions on Power Systems* 26.1 (2010), pp. 12–19.

[142] E. Zio. *Monte Carlo Simulation: The Method*. Springer, 2013.

[143] E. Zio and N. Pedroni. "Reliability analysis of discrete multi-state systems by means of subset simulation". In: *Proceedings of the 17th ESREL Conference*. 2008, pp. 22–25.

[144] K. M. Zuev, S. Wu, and J. L. Beck. "General network reliability problem and its efficient solution by subset simulation". In: *Probabilistic Engineering Mechanics* 40 (2015), pp. 25–35.

[145] M. J. Zuo, Z. Tian, and H.-Z. Huang. "An efficient method for reliability evaluation of multistate networks given all minimal path vectors". In: *IIE Transactions* 39.8 (2007), pp. 811–817.

[146] K. Zwirglmaier, J. Chan, I. Papaioannou, J. Song, and D. Straub. "Hybrid Bayesian networks for reliability assessment of infrastructure systems". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* (2023).

Part II

Published Papers

Chapter 4

An adaptive subset simulation algorithm for system reliability analysis with discontinuous limit states

Original Publication

J. Chan, I. Papaioannou, and D. Straub. "An adaptive subset simulation for system reliability analysis with discontinuous limit states". In: *Reliability Engineering & System Safety* 225 (2022), p. 108607.

Abstract

Many system reliability problems involve performance functions with a discontinuous distribution. Such situations occur in both connectivity- and flow-based network reliability problems due to binary or multi-state random variables entering the definition of the system performance or due to the discontinuous nature of the system model. When solving this kind of problem, the standard subset simulation algorithm with fixed intermediate conditional probability and fixed number of samples per level can lead to substantial errors since the discontinuity of the output can result in an ambiguous definition of the sought percentile of the samples and, hence, of the intermediate domains. In this paper, we propose an adaptive subset simulation algorithm to determine the reliability of systems whose performance function is a discontinuous random variable. The proposed algorithm chooses the number of samples and the intermediate conditional probabilities adaptively. We discuss two MCMC algorithms for the generation of the samples in the intermediate domains: the adaptive conditional sampling method and a novel independent Metropolis-Hastings algorithm that efficiently samples in discrete input spaces. The accuracy and efficiency of the proposed algorithm are demonstrated by a set of numerical examples.

4.1 Introduction

Infrastructure networks, such as power grids and water supply systems, deliver essential services to society. Failures of such networks can have severe consequences. Quantification of the probability of survival or, conversely, the probability of failure of such systems is essential in understanding and managing their reliability; this is the main purpose of network system reliability assessment.

For reliability analysis purposes, the performance of the system can be assessed by the limit state function (LSF), also known as performance function or structure function, $g(\boldsymbol{X})$. $\boldsymbol{X}$ is an n-dimensional vector of random variables with joint cumulative distribution function (CDF) $F_{\boldsymbol{X}}$ and represents the uncertainty in the model input. By convention, failure of the system occurs for all system states $\boldsymbol{x}$ for which $g(\boldsymbol{x}) \leqslant 0$. The probability of failure of the system is defined as

$$p_f \triangleq \mathbb{P}(g(\boldsymbol{X}) \leqslant 0) = \int_{g(\boldsymbol{x})\leqslant 0} \mathrm{d}F_{\boldsymbol{X}}(\boldsymbol{x}) \tag{4.1}$$

The vector of basic random variables $\boldsymbol{X}$ entering the definition of the LSF of network systems usually contains discrete random variables, which results in an LSF with discontinuous distribution. This is due to the fact that the performance of the network is often calculated through a function of a large number of binary or multi-state components. Moreover, real-world infrastructure networks are often designed to be highly reliable. This leads to high-dimensional reliability assessment problems with small failure probabilities [42].

Network performance is often measured through connectivity or 'travel time'(or flow) [14]. In connectivity-based problems, one evaluates the probability that a given set of nodes is connected, given that each component of the network fails with some probability. Typically, both the system performance and the component state are modeled as binary random variables. In this context, $g(\boldsymbol{X})$ is known as the structure function [32]. A set of sampling-based methods have been proposed for such kind of problems (e.g., [17, 16, 12, 25, 26, 27, 8, 37]), and a comparative study can be found in [34, 30].

In this paper, we focus on flow-based problems where the system performance and/or the component are typically modeled as multi-state or continuous random variables instead of binary ones, and, hence, most of the sampling techniques tailored for connectivity-based problems cannot be implemented directly. One of the major concerns in this area is the maximum flow that a stochastic network can deliver, i.e., the probability that the maximum flow from one or more source nodes to one or more terminal nodes is less than a predefined demand level. A number of sampling-based methods have been proposed for this type of problem [19, 18, 20, 1, 10, 31, 21, 13, 41]. However, all these methods assume that the edge capacities are independent and discrete random variables, which is often unrealistic. [42] employs the standard subset simulation (SuS) algorithm [4] to efficiently solve maximum-flow reliability problems, where both the edge capacity and the network performance are modeled as continuous random variables. [40] use the standard SuS algorithm in the reliability analysis of gas pipelines. However, as discussed in this paper, the adaptive approach of the standard SuS for determining the intermediate levels is not suitable for LSFs with discontinuous distribution, which is the case for most network reliability problems. As an example, Fig. 4.1 shows the CDF of the LSF of the IEEE39 bus benchmark system (described in Section 5.4). The CDF is discontinuous with 'jumps'. To overcome this limitation, one may construct a problem equivalent to the original one but with LSFs with continuous distribution and then use SuS to solve this equivalent problem.

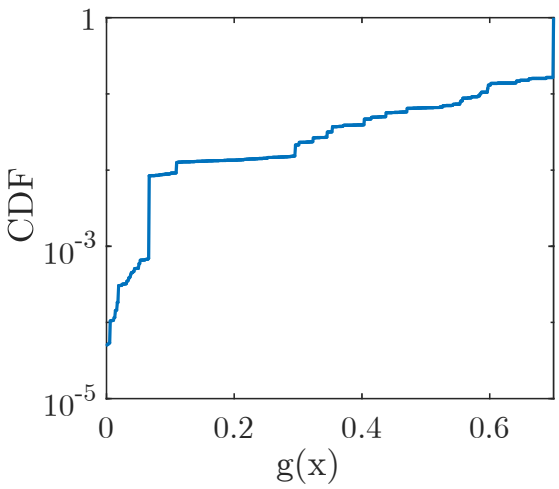


Figure 4.1: CDF of the LSF of the IEEE39 bus benchmark system of Section 5.4

This approach has been explored by Ching and Hsu [14] for connectivity-based problems, where a virtual random walk model is solved to get a continuous proxy of the original binary connectivity. Typically, such transformations need to be derived for the problem at hand.

The generalized splitting method [7] has also been employed to solve both connectivity- [8] and flow-based problems [9] in combination with tailored and efficient Gibbs samplers. It should be stressed that the determination of the intermediate levels in the generalized splitting method is through a pilot run of the adaptive multilevel splitting algorithm [7, 8, 6], which is essentially the standard SuS algorithm. Therefore, the transformation of discontinuously distributed LSF to a continuous one is also needed for these approaches.

The basic idea of SuS is to express the probability of failure as a product of larger conditional probabilities of a set of intermediate nested events. Two ingredients of the SuS algorithm are essential for obtaining an accurate and efficient estimator. The first is the efficient simulation of conditional samples, which is achieved through Markov Chain Monte Carlo (MCMC) methods [4, 28]. The second is the proper choice of the intermediate events. Similarly to the cross entropy method [29, 35], the intermediate failure events in SuS are chosen adaptively so that the estimates of the conditional probabilities equal a predefined value p_0. This is achieved through generating a fixed number of samples in each conditional level, sorting the samples according to their LSF values, and determining the p_0-percentile of the samples, which is set as the threshold defining the next intermediate failure event. When solving network reliability problems, the discontinuous nature of the LSF can result in a large number of samples in a certain conditional level having the same LSF value. In such cases, the standard SuS method will result in an ambiguous definition of the intermediate domains. In extreme conditions, all samples generated at a certain level might have the same LSF value, in which case the sample process can get stuck and might not reach the failure domain.

To address this issue, we introduce a novel variant of SuS called *adaptive effort subset simulation* (aE-SuS) method. Our method chooses the number of samples per level and the respective conditional probability adaptively to ensure that an adequate number of samples fall in the subsequent intermediate domain. Compared with other non-sampling based methods (e.g., [5, 11, 24, 39, 23]),

the proposed method facilities using advanced deterministic network analysis algorithms considering complex network dynamics like cascading failure. On the other hand, owing to its sampling nature, the aE-SuS algorithm may require a large number of simulations to achieve an acceptable result. It should be stressed that the proposed aE-SuS algorithm is applicable for dependent input random variables, and any MCMC algorithm that enables efficient sampling of the intermediate conditional distributions can be combined with the proposed algorithm.

The paper is organized as follows: Section 2 gives a brief introduction to the standard SuS. Section 3 discusses two MCMC algorithms in the context of network reliability assessment. Section 4 introduces the basic idea as well as the implementation details of the aE-SuS method. In Section 5, the performance of the proposed algorithm is illustrated by a set of numerical examples, a one-dimensional multi-state problem, a multidimensional flow-based problem with combined continuous and binary capacities, a binomial experiment with small success probability, and a benchmark power transmission network system. The paper closes with the conclusions in Section 6.

4.2 Standard subset simulation

4.2.1 Brief introduction of subset simulation

The basic idea of SuS is to express the rare failure event $F = \{\boldsymbol{x} : g(\boldsymbol{x} \leqslant 0)\}$ as the intersection of a sequence of nested intermediate events $F_1 \supset F_2 \supset \cdots \supset F_m$. Owing to the nestedness of the intermediate events, the failure event can be expressed as $F = \cap_{l=1}^{m} F_l$. The failure probability can then be decomposed as the following product of conditional probabilities:

$$\mathbb{P}(F) = \prod_{l=1}^{m} \mathbb{P}(F_l | F_{l-1}) \tag{4.2}$$

where F_0 is the certain event. Ideally, the intermediate events are selected such that each conditional probability is large, typically $\geqslant 0.1$. In this way, the original problem of estimating a small probability is transformed to a sequence of m intermediate problems of evaluating larger conditional probabilities.

The estimation of each conditional probability $\mathbb{P}(F_l|F_{l-1})$ requires sampling from the distribution of the random variables conditional on F_{l-1}, denoted as $Q(\cdot|F_{l-1})$, where $Q(\cdot|F_0)$ represents the initial input distribution and equals the generalized derivative of the input CDF $F_{\boldsymbol{X}}(\cdot)$. $Q(\cdot|F_0)$ can be sampled by standard Monte Carlo sampling, but the distributions $Q(\cdot|F_l), l > 0$, are only known point-wise up to a normalizing constant and, hence, cannot be sampled directly. Therefore, MCMC sampling is employed. The sampling process in the l-th conditional sampling level is performed as follows: (1) Select the samples $\mathcal{P}^{(l-1)}$ from the $(l-1)$-th level that fall in F_l as the seeds $\mathcal{S}^{(l)}$ ($\mathcal{P}^{(0)}$ is generated through Monte Carlo sampling). (2) From each seed, start a Markov chain that has the target distribution $Q(\cdot|F_l)$ as the stationary distribution, and record all the states as new samples $\mathcal{P}^{(l)}$. (3) Take the samples $\mathcal{P}^{(l)}$ located in F_{l+1} as new seeds $\mathcal{S}^{(l+1)}$ and estimate $\mathbb{P}(F_{l+1}|F_l)$ as $\frac{|\mathcal{S}^{(l+1)}|}{|\mathcal{P}^{(l)}|}$ where $|\mathcal{S}^{(l+1)}|$ and $|\mathcal{P}^{(l)}|$ denote the number of seeds and samples, respectively. The above three steps are repeated successively until F is approached. We note that the number of samples per level $|\mathcal{P}^{(l)}|$ is usually fixed prior to the analysis.

Defining the intermediate events a priori is typically challenging. Hence, in standard SuS the intermediate failure events are chosen adaptively during the simulation such that each conditional probability equals a predefined constant p_0. This standard SuS approach is also termed (fixed effort) adaptive multilevel splitting [7]. In this variant, step (3) in the above sampling process is modified as follows: Order the samples $\mathcal{P}^{(l)}$ by their LSF values. The first p_0-percent of these sorted samples are then taken as seeds for the next sampling level and the LSF value of the p_0-percentile b_{l+1} is used to define the boundary of the next intermediate domain, such that $F_{l+1} = \{\boldsymbol{x} : g(\boldsymbol{x}) \leqslant b_{l+1}\}$. The resulting SuS estimator of the probability of failure is given as:

$$\widehat{p}_f = p_0^{m-1} \frac{N_f}{N} \tag{4.3}$$

where N and N_f represent the number of samples and failure samples at the final level, respectively. The standard SuS algorithm is summarized in Algorithm 1.

Algorithm 1: SuS algorithm

Input: $p_0 \in (0,1)$, an integer N multiple of $\frac{1}{p_0}$

$l \leftarrow 0$, $b_l \leftarrow \inf$
while $b_l > 0$ **do**
  if $l = 0$ **then**
    Generate N samples $\{\boldsymbol{x}_k\}_{k=1}^N$ from the initial distribution $Q(\cdot|F_0)$
  else
    Generate N samples $\{\boldsymbol{x}_k\}_{k=1}^N$ from the target distribution $Q(\cdot|F_l)$ with an MCMC algorithm with seeds $\mathcal{S}^{(l)}$
  Sort $\{\boldsymbol{x}_k\}_{k=1}^N$ by increasing order of their LSFs $g(\cdot)$, and denote the sorted samples as $\{\bar{\boldsymbol{x}}_k\}_{k=1}^N$
  $b_{l+1} \leftarrow g(\bar{\boldsymbol{x}}_{p_0 \cdot N})$
  if $b_{l+1} \leqslant 0$ **then**
    $b_{l+1} \leftarrow 0$
    $N_f = \sum_{k=1}^N \mathbb{I}\{g(\bar{\boldsymbol{x}}_k) \leqslant 0\}$
  Take the $\mathcal{S}^{(l+1)} \triangleq \{\bar{\boldsymbol{x}}_k\}_{k=1}^{p_0 \cdot N}$ as the seeds for the next level
  $l \leftarrow l + 1$
$\widehat{p}_f \leftarrow p_0^{l-1} \frac{N_f}{N}$

Output: $\widehat{p}_f$

As previously mentioned, MCMC sampling is applied to generate samples from each conditional distribution $Q(\cdot|F_l)$. In SuS, the seeds $\mathcal{S}^{(l)}$ already follow approximately the target distribution [4]; hence, a burn-in period is not considered in practice. Since the samples generated from the same seed are states of the same Markov chain, they will be dependent; their correlation depends on the autocorrelation function of the underlying Markov chain. The stronger the correlation between samples, the larger the variance of the estimates of the conditional probabilities. Additionally, samples that share the same history, namely, their Markov chains are branches with roots at the same Monte Carlo sample, will be correlated, even when they are at different levels. Such correlation introduces a dependency on the conditional probability estimators, which further increases the variance of the SuS estimator. Hence, the quality of the final probability estimate strongly depends on the particular choice and setting of the MCMC algorithm.

4.2.2 Accuracy of the Subset Simulation estimator

The accuracy of the SuS estimator of the probability of failure $\widehat{p}_f$ can be assessed by the mean-square error, which is decomposed as:

$$\mathrm{MSE}(\widehat{p}_f) = (p_f - \mathbb{E}(\widehat{p}_f))^2 + \mathrm{Var}(\widehat{p}_f) \tag{4.4}$$

The first term on the right-hand side of Eq. (4.4) represents the bias contribution, and the second term is the variance of the SuS estimator.

Assume first that the intermediate events are defined before the simulation. In the Monte Carlo level ($l = 0$), samples $\mathcal{P}^{(0)}$ are generated from $Q(\cdot|F_0)$ independently, and therefore the seeds $\mathcal{S}^{(1)}$ follow the distribution $Q(\cdot|F_1)$. This will lead to so-called perfect sampling when simulating the Markov chains in the next level. Since the chains have already reached the stationary state at the beginning, no burn-in time is needed, and all samples $\mathcal{P}^{(1)}$ will follow $Q(\cdot|F_1)$. In this way, samples $\mathcal{P}^{(l)}$ generated in any l-th conditional level will follow the target distribution $Q(\cdot|F_l)$ and the corresponding estimator of the conditional probability $\widehat{p}(F_{l+1}|F_l)$ will be unbiased. Moreover, [7] proves that the resulting failure probability estimator $\widehat{p}_f$ is also unbiased if both intermediate events and length of the Markov chain are predefined, i.e., if they are independent of the simulation process.

Since the intermediate events are selected adaptively in SuS, samples $\mathcal{S}^{(l)}$ will not completely follow the target distribution. As a result, both the conditional probability estimator and failure probability estimator will be slightly biased. Nevertheless, compared to the variance of the estimator, the squared bias is one order of magnitude smaller [4] and, hence, its contribution to the mean-square error (MSE) of the estimator is negligible. In other words, the error of the SuS is mainly due to the variance of the failure probability estimator rather than the bias. The most common and reliable way to calculate the variance $\mathrm{Var}(\widehat{p}_f)$ is to run SuS several times and to use the sample variance as the unbiased estimation of the $\mathrm{Var}(\widehat{p}_f)$. One can also evaluate the variance approximately through a single run of the SuS. More details can be found in [4] and [28]. However, this approximate estimator is shown to underrepresent the true variance of $\widehat{p}_f$, especially for small target p_f.

4.3 Markov Chain Monte Carlo algorithm for network reliability assessment

Most MCMC algorithms that are widely used in risk analysis can be regarded as variants of the Metropolis-Hastings (M-H) algorithm. These include, for example, Gibbs sampling [8][9], conditional sampling [28] and Hamiltonian Monte Carlo [38]. To sample from the intermediate target distribution $Q(\cdot|F_l)$ in SuS, M-H algorithm proceeds in the following two steps [28]:

1. Generate a candidate sample $\boldsymbol{v}$ from the distribution $Q_p^{(l)}(\cdot|\boldsymbol{x}_k)$ which is termed the proposal distribution.

2. Accept or reject $\boldsymbol{v}$.

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{v}, & \text{with prob.} \quad \alpha \\ \boldsymbol{x}_k, & \text{with prob.} \quad 1-\alpha \end{cases} \quad \text{where}$$
$$\alpha = \mathbb{I}\{\boldsymbol{v} \in F_l\} \min\left\{1, \frac{Q(\boldsymbol{v}|F_0)Q_p^{(l)}(\boldsymbol{x}_k|\boldsymbol{v})}{Q(\boldsymbol{x}_k|F_0)Q_p^{(l)}(\boldsymbol{v}|\boldsymbol{x}_k)}\right\}$$

It can be shown that the M-H algorithm satisfies the detailed balance condition independent of the choice of the proposal distribution. In this section, we first discuss the adaptive conditional sampling method of [28] in the context of network reliability assessment and then propose a more efficient yet less general independent M-H algorithm, which is applicable in problems with discrete input spaces.

4.3.1 Adaptive conditional sampling in standard normal space

4.3.1.1 Implementation in standard normal space

Let $\boldsymbol{U}$ denote an n-dimensional random vector that has the independent standard normal distribution. The original random vector $\boldsymbol{X}$ can be expressed in terms of the vector $\boldsymbol{U}$ through an isoprobabilistic mapping $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^n$. One can define the reliability problem in the $\boldsymbol{U}$-space as follows:

$$p_f = \mathbb{P}(g(\boldsymbol{X}) \leqslant 0) = \mathbb{P}(G(\boldsymbol{U}) \leqslant 0) = \int_{G(\boldsymbol{u}\leqslant 0)} \varphi_n(\boldsymbol{u}) \mathrm{d}\boldsymbol{u} \tag{4.5}$$

where $G(\boldsymbol{U}) = g(\boldsymbol{T}(\boldsymbol{U}))$ and $\varphi_n(\boldsymbol{u})$ is the n-dimensional independent standard normal joint probability density function (PDF). The mapping $\boldsymbol{T}(\cdot)$ can be obtained by the Rosenblatt transformation, which is implemented as follows:

$$\begin{aligned} x_1 &= F_{X_1}^{-1}(\Phi(u_1)) \\ x_2 &= F_{X_2}^{-1}(\Phi(u_2)|x_1) \\ &\vdots \\ x_n &= F_{X_n}^{-1}(\Phi(u_m)|x_1, \cdots, x_{n-1}) \end{aligned} \tag{4.6}$$

where Φ represents the CDF of standard normal distribution and $F_{X_d}(\cdot|x_1, \cdots, x_{d-1})$ denotes the conditional CDF of X_d given $X_1 = x_1, \cdots, X_{d-1} = x_{d-1}$. If any subset of $\boldsymbol{X}$ consists of discrete random variables, then it is possible that the functions $F_{X_d}(\cdot|x_1, \cdots, x_{d-1})$ are not strictly invertible. Therefore, we use the following extended definition of the inverse of a CDF

$$F^{-1}(a) = \inf(x : F(x) \geq a) \tag{4.7}$$

We note that, in such cases, the Rosenblatt transformation is not one-to-one, and hence, the inverse mapping from $\boldsymbol{X}$ to $\boldsymbol{U}$ is not uniquely defined.

4.3.1.2 Adaptive conditional sampling algorithm

Having defined the transformation from the original $\boldsymbol{X}$-space to the $\boldsymbol{U}$-space, SuS can be used to solve the reliability problem in the transformed space. At the l-th level of SuS, MCMC sampling is

applied to sample from the conditional standard normal density $p_{\boldsymbol{U}}(\cdot|F_l)$. Sampling according to this density can be performed by application of the aCS algorithm. Before describing the aCS algorithm, we first discuss its non-adaptive variant, the standard conditional sampling (CS) algorithm. The transition from the current state $\boldsymbol{u}_k$ to a new state $\boldsymbol{u}_{k+1}$ using CS is as follows: First, a candidate $\boldsymbol{v}$ is generated. The CS sampler imposes that the candidate and the current state are jointly Gaussian with standard normal marginal distribution $\varphi_n(\cdot)$ and predefined symmetric cross-correlation matrix, $\boldsymbol{R}$. One then samples $\boldsymbol{v}$ from the joint Gaussian distribution conditional on the current state, i.e., from $\mathcal{N}(\boldsymbol{v}; \boldsymbol{R}\boldsymbol{u}_k, \boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}})$, where $\boldsymbol{I}$ is the identity matrix. The candidate is accepted if it is located in the intermediate failure domain F_l, in which case it is set as the new state $\boldsymbol{u}_{k+1}$. Otherwise, $\boldsymbol{u}_k$ is taken as the new state. CS can be summarized as follows:

1. Generate candidate sample $\boldsymbol{v}$ from the normal distribution $\mathcal{N}(\boldsymbol{v}; \boldsymbol{R}\boldsymbol{u}_k, \boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}})$.

2. Accept or reject $\boldsymbol{v}$.
$$\boldsymbol{u}_{k+1} = \begin{cases} \boldsymbol{v}, \boldsymbol{v} \in F_l \\ \boldsymbol{u}_k, \boldsymbol{v} \notin F_l \end{cases}$$

It can be proven that the above transition satisfies the detailed balance condition with respect to the target distribution $p_{\boldsymbol{U}}(\cdot|F_l)$, hence, $p_{\boldsymbol{U}}(\cdot|F_l)$ is the stationary distribution of the generated Markov chain [28, 3]. In fact, CS can be regarded as the M-H sampler with proposal distribution taken as $\mathcal{N}(\boldsymbol{v}; \boldsymbol{R}\boldsymbol{u}_k, \boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}})$ [28]. We further note that the CS sampler will never generate repeated candidates, which results in an acceptance probability that is independent of the dimension of the vector $\boldsymbol{U}$. Hence, it is suitable for application to high-dimensional problems.

The performance of the CS sampler depends on the choice of the matrix $\boldsymbol{R}$ or, equivalently, the covariance matrix of the proposal distribution $\mathcal{N}$. Usually, $\boldsymbol{R}$ is chosen as a diagonal matrix with d-th diagonal term equal to ρ_d, which implies a component-wise sampling scheme, i.e., the d-th component of candidate, v_d, is sampled from $\mathcal{N}(v_d; \rho_d u_{k,d}, 1-\rho_d^2)$. Large values of ρ_d lead to a strong correlation between the current and the next state, but small values also lead to increased correlation due to the high rejection rate in the second step of CS. The correlation of the generated Markov chains can be controlled by choosing ρ_d or, equivalently, the standard deviation $\sigma_d = \sqrt{1-\rho_d^2}$ adaptively, employing intermediate results from the simulation. In adaptive MCMC algorithms, the chain correlation is usually controlled by matching a near-optimal acceptance probability of the chain [2]. The aCS algorithm [28] adapts the sampling parameters by running batches of N_a chains starting from randomly selected seeds. After running each batch, the acceptance probability is estimated, and the sampling parameters are adapted to match a pre-defined acceptance probability α^*. The aCS algorithm for generating N samples according to $p_{\boldsymbol{U}}(\cdot|F_l)$ starting from seeds $\mathcal{S}^{(l)}$ is given in Appendix 4.A.

4.3.2 Independent Metropolis-Hastings algorithm

In network reliability assessment, the probability content at the intermediate domains in SuS typically centers at multiple discrete system states (modes), and hence, the intermediate target distribution is multimodal. To efficiently sample from such distribution, we propose a novel independent

M-H algorithm. The algorithm is applicable when all input random variables are discrete and exploits the information of the discarded samples in previous sampling levels to form a proper proposal distribution in the M-H algorithm that is independent of the current state of the chain. Specifically, let $\mathcal{X}_l$ represent the set of samples discarded at level l, in other words, the generated samples at level l that are not in the $(l+1)$-th intermediate domain. The proposal distribution $Q_p^{(l)}(\boldsymbol{x})$ of the M-H algorithm at level l is then defined as the original input distribution $Q(\boldsymbol{x}|F_0)$ excluding the states visited by the previously discarded samples. That is,

$$Q_p^{(l)}(\boldsymbol{x}) \propto Q(\boldsymbol{x}|F_0)\mathbb{I}\{\boldsymbol{x} \notin \cup_{i=0,\dots,l-1}\mathcal{X}_i\} \tag{4.8}$$

Since all samples in $\cup_{i=0,\dots,l-1}\mathcal{X}_i$ are located outside the intermediate domain F_l, F_l is included in the support of the proposal distribution, $\Omega_p^{(l)} = \boldsymbol{x} \notin \cup_{i=0,\dots,l-1}\mathcal{X}_i$. This is illustrated in Fig. 4.2. Moreover, the proposal distribution has exactly the same shape as the intermediate target

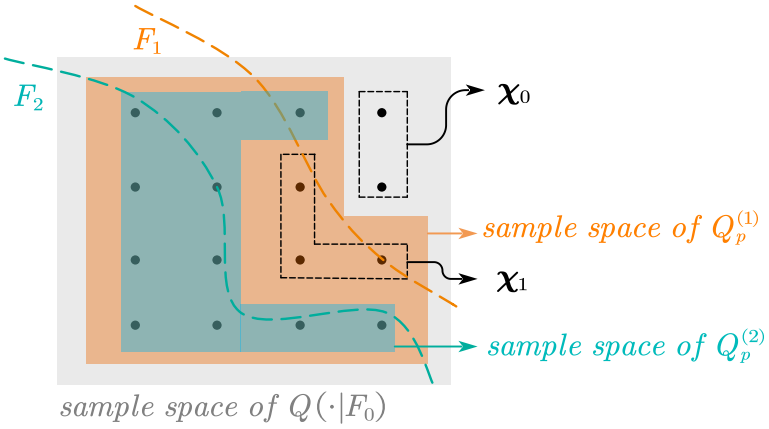


Figure 4.2: Schematic diagram of proposal distributions. (The black dots represent the basic random events, and the dotted curve indicates the intermediate failure domain)

distribution in F_l as they are both proportional to the input distribution $Q(\boldsymbol{x}|F_0)$. The acceptance rate α of the candidate generated by this proposal is given as

$$\alpha = \mathbb{I}\{\boldsymbol{x} \in F_l\} \min\left\{1, \frac{Q(\boldsymbol{x}|F_0)Q(\boldsymbol{x}_k|F_0)}{Q(\boldsymbol{x}_k|F_0)Q(\boldsymbol{x}|F_0)}\right\} = \mathbb{I}\{\boldsymbol{x} \in F_l\} \tag{4.9}$$

Note that both the generation and the acceptance of the candidate are independent of the current state. In the literature, M-H samplers whose proposal distribution is independent of the current state are termed independent M-H samplers. We note that the samples generated by this algorithm are not independent since we get a repeated sample when rejecting the candidate. This is the main difference between the independent M-H algorithm and rejection sampling with envelope $\frac{1}{\mathbb{E}[\alpha]}Q_p^{(l)}(\boldsymbol{x})$, which generates independent samples [22]. In the latter, one gets a new sample only when the candidate is accepted and, hence, the computational cost of rejection sampling is much higher than the independent M-H algorithm, especially as the mean acceptance rate $\mathbb{E}[\alpha]$ is small. The average acceptance probability, $\mathbb{E}[\alpha]$, can be calculated through dividing $\mathbb{P}(\boldsymbol{X} \in F_l)$ by $\mathbb{P}(\boldsymbol{X} \notin \cup_{i=0,\dots,l-1}\mathcal{X}_i)$, i.e.,

$$\mathbb{E}_{Q_p^{(l)}}[\alpha] = \frac{\mathbb{P}(\boldsymbol{X} \in F_l)}{\mathbb{P}(\boldsymbol{X} \notin \cup_{i=0,\dots,l-1}\mathcal{X}_i)} \tag{4.10}$$

The magnitude of $\mathbb{E}[\alpha]$ tends to decrease as the intermediate level goes higher. Additionally, the sample size at each level, the dimension of the problem, and the input distribution will also influence

the mean acceptance rate $\mathbb{E}[\alpha]$. The influence is investigated in detail through a binomial experiment in Section 4.5.

The implementation of the above independent M-H algorithm is relatively simple and can be performed in the following two steps:

1. Generate candidate sample $\boldsymbol{v}$ from the proposal distribution $Q_p^{(l)}(\boldsymbol{x})$.
2. Accept or reject $\boldsymbol{v}$.
$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{v}, \boldsymbol{v} \in F_l \\ \boldsymbol{x}_k, \boldsymbol{v} \notin F_l \end{cases}$$

To sample from the proposal distribution, $Q_p^{(l)}(\boldsymbol{x})$, one can sample directly from the input distribution and keep those samples that differ from the previous discarded samples. However, such a process can be quite inefficient when the proposal distribution is far from the input distribution, for instance, at deep intermediate levels. Such an issue can be circumvented by applying the bound-based sampling algorithm [34], given that the input random variables are multivariate categorical distributed. Details on this algorithm can be found in Appendix 4.B.

4.4 Adaptive effort subset simulation method

In each conditional level l of the SuS method with a fixed number of samples per level and adaptive estimation of the intermediate events, the p_0-percentile of the LSF values of the samples $\mathcal{P}^{(l)}$, b_{l+1}, is used to define the boundary of the intermediate domain. This adaptive approach works well when only a few samples are located on the boundary $g(\boldsymbol{x}) = b_{l+1}$, i.e., a few samples have the same LSF value as the p_0-percentile. However, it can happen that many samples fall on this boundary, particularly in the following cases:

(1) $\boldsymbol{X}$ includes discrete random variables.

(2) The LSF is defined such that the probability measure of the set $\{\boldsymbol{x} : g(\boldsymbol{x}) = b_{l+1}\}$ is strictly greater than zero.

(3) The parameters of the MCMC algorithm are inappropriately set, resulting in the candidates being rejected successively many times.

While case (3) can be avoided by an appropriate implementation of the MCMC algorithm, cases (1) and (2) are common in the context of network reliability assessment. This will result in an ambiguous definition of the intermediate domain F_{l+1} and can lead to an inaccurate estimate of the failure probability. In extreme situations, all samples generated in a certain level will have the same LSF value and the adaptive sampling process can get stuck and never reach the failure domain.

In this section, we modify the standard SuS algorithm to circumvent this problem. The resulting algorithm modifies the adaptive selection of the intermediate domains and adapts the number of

samples per level (sampling effort) throughout the simulation. We term the proposed approach aE-SuS. As will be made clear, these modifications enable the application of the method to general network reliability problems. The proposed algorithm is introduced in the following and summarized in Algorithm 2.

4.4.1 Intermediate domains

In order to provide a clear (unambiguous) definition of the intermediate domains, one can apply the following adaptive approach. At each conditional level, generate a set of samples $\mathcal{P}^{(l)}$ and define a temporary event F_{temp} as $\{\boldsymbol{x} : g(\boldsymbol{x}) \leqslant b_{l+1}\}$ where b_{l+1} is the p_0-percentile of the LSF values of $\mathcal{P}^{(l)}$. If $F_{temp} = F_l$, define the next intermediate event F_{l+1} as $\{\boldsymbol{x} : g(\boldsymbol{x}) < b_{l+1}\}$, otherwise set $F_{l+1} = F_{temp}$. This approach guarantees that $F_{l+1} \subsetneq F_l$, which avoids a degeneracy of the sampling process.

Because of the discrete nature of $g(\boldsymbol{x})$, it might be difficult to check whether $F_{temp} = F_l$ or not when $F_l = \{\boldsymbol{x} : g(\boldsymbol{x}) < b_l\}$. Therefore, we check if $b_{l+1} = \max\{g(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{P}^{(l)}\}$ instead. The latter condition checks whether the p_0 percentile of the LSF values equals the maximum LSF value of the samples $\mathcal{P}^{(l)}$, and is a necessary (but not sufficient) condition of $F_{temp} = F_l$. Note that $F_{l+1} \subsetneq F_l$ still remains true after this modification, since $F_l / F_{l+1} \neq \emptyset$ and contains at least the samples taking the maximum LSF value in $\mathcal{P}^{(l)}$. The above adaptive approach for choosing the intermediate domains is described in lines 11-20 of Algorithm 2.

4.4.2 Sampling at the intermediate levels

The approach for selecting the intermediate domains, introduced in Section 3.1, could potentially lead to a very small number of failure samples per level, which reduces the accuracy of the estimates of the intermediate conditional probabilities. We hence need to adapt the number of samples per level to ensure that these estimates remain accurate. We first calculate the number of samples that fall into the domain F_{l+1} (the number of seeds N_s). If this number is smaller than a predefined constant C, we increase the current sampling effort N_c and append $\mathcal{P}^{(l)}$ with N new samples. Denote the extended sample set as $\mathcal{P}^{(l)}_{\text{ext}}$. The new samples should also follow the target distribution $Q(\cdot|F_l)$, and hence approximately $\mathbb{P}(F_{l+1}|F_l)$ of these samples should be located in F_{l+1}. Therefore, one needs to further generate $N = N_c \cdot \frac{C - N_s}{N_s}$ samples to get approximately C seeds, which is shown in line 22 of Algorithm 2. Note that $N > 0$ always holds. In the algorithm, the constant C is taken as a predefined proportion $tol \in (0, 1)$ of $p_0 \cdot N_0$, the product of initial conditional probability and the initial sample size. Larger tol will lead to more accurate but less efficient results. We have found $tol \in (0.5, 0.8)$ to be a good choice for the investigated cases.

In practice, the above appending process may need to be iterated several times to achieve at least C seeds. For a fixed F_{l+1}, with every iteration and increasing number of samples, the number of the seeds will keep increasing until the desired threshold C is achieved. By doing this, even in the extreme case where all the samples in $\mathcal{P}^{(l)}$ have the same LSF value, the sampling process will keep moving forward towards the failure domain and will no longer get stuck in this level as in the standard SuS algorithm.

To append new samples that follow the target distribution $Q(\cdot|F_l)$, we propose to extend the Markov chains generated in the initial intermediate sampling step ($iter = 0$, in Algorithm 2). This is illustrated in Fig. 4.3. As shown in the figure, for each seed in $\mathcal{S}^{(l)}$, a Markov chain is constructed in the 0th iteration. The last sample (tail) of this chain is then taken as the seed for the chain in the next iteration. The transition distribution of the chain remains unchanged. The above process may be iterated several times and is described as Algorithm 3. For iteration $it = 0, \cdots$, the input of Algorithm 3 consists of 4 values: the number of samples to append N, the number of Markov chains $N_{ch} = |\mathcal{S}^{(l)}|$, the transition distribution of each chain $\{\Gamma_i\}_{i=1}^{N_{ch}}$, which is determined by the target distribution $Q(\cdot|F_l)$ and the employed MCMC algorithm, and the seed for each chain $\{\boldsymbol{e}_i^{(it)}\}_{i=1}^{N_{ch}}$, which is taken as $\mathcal{S}^{(l)}$ when $it = 0$ and otherwise as the tail of the Markov chains from the previous iteration $\{\boldsymbol{t}_i^{(it-1)}\}_{i=1}^{N_{ch}}$. The output of the algorithm is N new generated samples $\mathcal{P}_{\text{new}}^{(l)}$ and the tail (last sample) of each chain $\{\boldsymbol{t}_i^{(it)}\}_{i=1}^{N_{ch}}$.

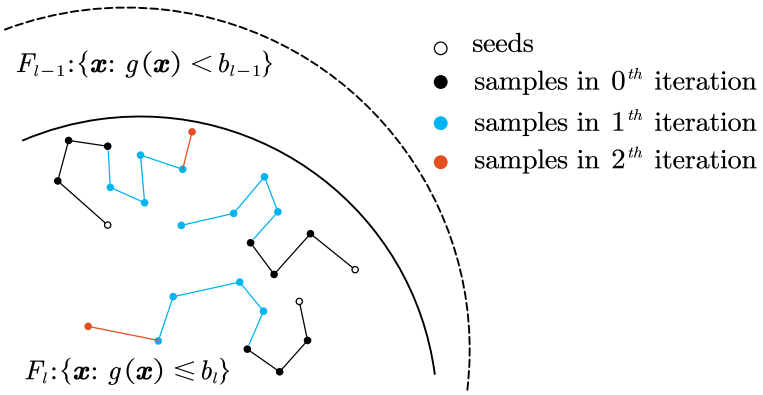


Figure 4.3: Schematic diagram of the appending method.

4.5 Examples

4.5.1 Multistate random variable

Consider a discrete random variable X with 7 states $\{x_1, \cdots, x_7\}$. We consider two cases. In case 1, the CDF of X, $F_X(\cdot)$, is set such that $\frac{F_X(x_{i+1})}{F_X(x_i)} \leq 10$, while in case 2, there is a big 'jump' between the third and the fourth state, i.e., $\frac{F_X(x_4)}{F_X(x_3)} \approx 599$. The CDF of X for the two considered cases is given in Table 4.1 and is illustrated in Fig 4.4. The LSF is defined as $g(X) = X + 5$ such that the failure probability $\mathbb{P}(X \leqslant -5)$ equals 10^{-5} for both cases.

We implement SuS and the proposed aE-SuS, respectively, in standard normal space to evaluate the failure probability and compare them with crude MCS. The MCMC algorithm is aCS. For SuS, the sampling effort is fixed to 1,000, and the conditional probability is 0.1. For aE-SuS, the parameters are set to be $tol = 0.5, N_0 = 1{,}000, p_0 = 0.1$. Each method is run 1,000 times to get the relative bias, coefficient of variation, and average computational cost of the failure probability estimator. The results for case 1 and case 2 are shown in Tables 4.2 and 4.3 respectively. In both cases, aE-SuS shows good accuracy, a negligible bias, and a much smaller coefficient of variation than crude MCS. We note that the coefficient of variation of crude MCS is given for the same computational effort

Algorithm 2: Adaptive effort subset simulation algorithm

Input: $tol \in (0,1)$, $p_0 \in (0,1)$, an integer N_0 multiple of $1/p_0$

$l \leftarrow 0, b_l \leftarrow \inf, N \leftarrow N_0, \mathcal{P}^{(l)} \leftarrow \emptyset$

while $b_l > 0$ **do**

  $iter \leftarrow 0$, $N_s \leftarrow 0$

  while $N_s < tol \cdot N_0 \cdot p_0$ **do**

    if $l = 0$ **then**

      Generate N samples $\{\boldsymbol{x}_k\}_{k=1}^{N}$ from the initial distribution $Q(\cdot|F_0)$ and add them to $\mathcal{P}^{(l)}$

    else

      Generate N samples $\{\boldsymbol{x}_k\}_{k=1}^{N}$ from the target distribution $Q(\cdot|F_l)$ with appending algorithm and add them to $\mathcal{P}^{(l)}$

    $N_c \leftarrow |\mathcal{P}^{(l)}|$ // total sample size

    Sort the elements of $\mathcal{P}^{(l)}$ by increasing order of their LSF values $g(\boldsymbol{x})$, and denote the sorted samples as $\{\bar{\boldsymbol{x}}_k\}_{k=1}^{N_c}$

    if $iter = 0$ **then**

      $b_{l+1} \leftarrow g(\bar{\boldsymbol{x}}_{p_0 \cdot N_0})$

      if $b_{l+1} \leqslant 0$ **then**

        $b_{l+1} \leftarrow 0$

        $N_s \leftarrow \sum_{k=1}^{N_c} \mathbb{I}\{g(\bar{\boldsymbol{x}}_k) \leqslant b_{l+1}\}$, $F_{l+1} \triangleq \{\boldsymbol{x} : g(\boldsymbol{x}) \leqslant b_{l+1}\}$

        Break

      else if $b_{l+1} < g(\bar{\boldsymbol{x}}_{N_c})$ **then**

        $N_s \leftarrow \sum_{k=1}^{N_c} \mathbb{I}\{g(\bar{\boldsymbol{x}}_k) \leqslant b_{l+1}\}$, $F_{l+1} \triangleq \{\boldsymbol{x} : g(\boldsymbol{x}) \leqslant b_{l+1}\}$

        Break

    $N_s \leftarrow \sum_{k=1}^{N_c} \mathbb{I}\{g(\bar{\boldsymbol{x}}_k) < b_{l+1}\}$, $F_{l+1} \triangleq \{\boldsymbol{x} : g(\boldsymbol{x}) < b_{l+1}\}$

    if $N_s < tol \cdot N_0 \cdot p_0$ **then**

      $N \leftarrow \lceil N_c \cdot \frac{tol \cdot N_0 \cdot p_0}{max(1, N_s)} \rceil - N_c \geqslant 1$

      $iter \leftarrow iter + 1$

  Take the $\mathcal{S}^{(l+1)} \triangleq \{\bar{\boldsymbol{x}}_k\}_{k=1}^{N_s}$ as the seeds for the next level

  $N \leftarrow N_0 - N_s$, $\mathcal{P}^{(l+1)} \leftarrow \mathcal{S}^{(l+1)}$, $l \leftarrow l + 1$

  $\widehat{p}(F_l|F_{l-1}) \leftarrow \frac{N_s}{N_c}$

$\widehat{p}(F) \leftarrow \prod_{j=1}^{l} \widehat{p}(F_j|F_{j-1})$

Output: $\widehat{p}(F)$

Algorithm 3: Appending algorithm

Input: N, N_{ch}, $\{\Gamma_i\}_{i=1}^{N_{ch}}$, $\{\boldsymbol{e}_i^{(it)}\}_{i=1}^{N_{ch}}$

Randomly choose $\mathrm{mod}(N, N_{ch})$ elements from the set $\{1, 2, \cdots, N_{ch}\}$, say $\boldsymbol{\chi}$

$\mathcal{P}_{\text{new}}^{(l)} \leftarrow \emptyset$

for $i = 1, \cdots, N_{ch}$ **do**

  if $i \in \boldsymbol{\chi}$ **then**

    $j \leftarrow \lfloor \frac{N}{N_{ch}} \rfloor + 1$

  else

    $j \leftarrow \lfloor \frac{N}{N_{ch}} \rfloor$

  $\boldsymbol{x}_0 \leftarrow \boldsymbol{e}_i^{(it)}$

  for $k = 1, \cdots, j$ **do**

    Sample $\boldsymbol{x}_k$ from transition density $\Gamma_i(\cdot|\boldsymbol{x}_{k-1})$

    Add $\boldsymbol{x}_k$ to $\mathcal{P}_{\text{new}}^{(l)}$

  $\boldsymbol{t}_i^{(it)} \leftarrow \boldsymbol{x}_j$

Output: $\mathcal{P}_{\text{new}}^{(l)}$, $\{\boldsymbol{t}_i^{(it)}\}_{i=1}^{N_{ch}}$

as the proposed aE-SuS method. In contrast, SuS gives a strongly biased estimate of the failure probability with a high coefficient of variation in the first case and falls into a dead loop in the second case.

Table 4.1: CDF of X for Example 5.1.

State	-6	-4	-3	-2	-1	0	1
CDF(case1)	1e-5	1e-4	1e-3	1e-2	1e-1	5e-1	1
CDF(case2)	1e-5	3e-5	5e-5	3e-2	1e-1	5e-1	1

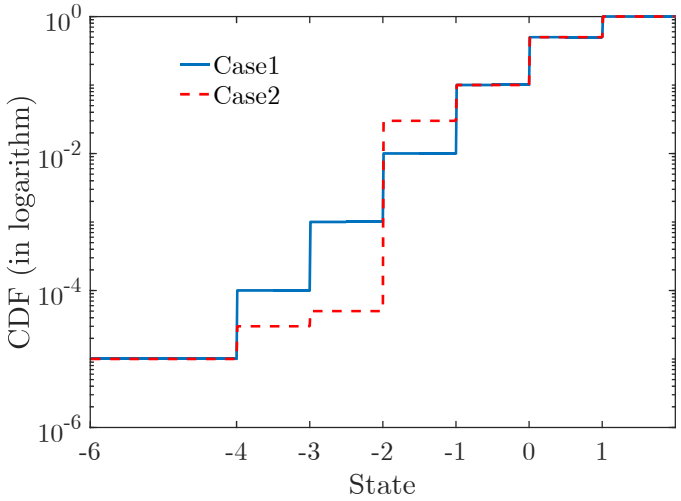


Figure 4.4: CDF of X for Example 5.1.

Table 4.2: Statistical characteristics of the estimator of the probability of Example 5.1 (Case 1).

	relative bias(%)	coefficient of variation	average computational effort
SuS	-97.8	3.747	7,222
aE-SuS	3.5	0.376	5,970
MCS	0	4.093	5,970

Table 4.3: Statistical characteristics of the estimator of the probability of Example 5.1 (Case 2).

	relative bias(%)	coefficient of variation	average computational effort
SuS	/	/	/
aE-SuS	2.4	0.242	44,737
MCS	0	1.495	44,737

4.5.2 Multidimensional flow-based problem

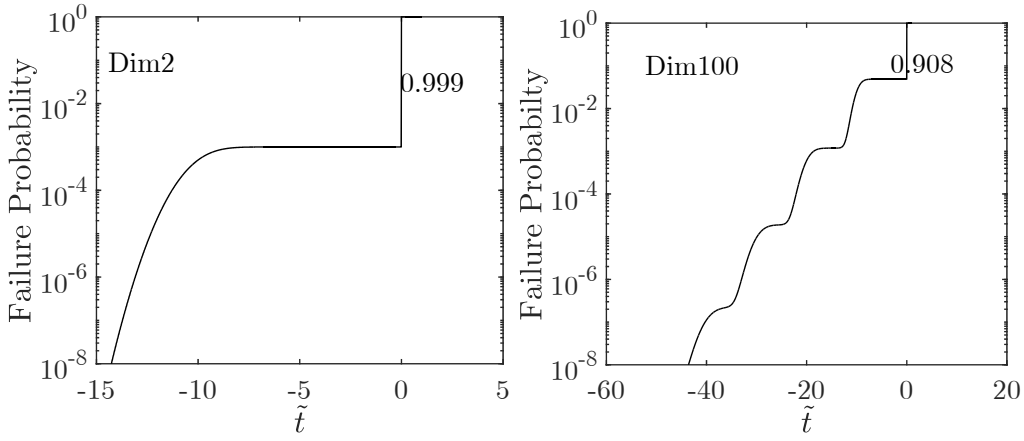
In this example, the failure event is defined as

$$\sum_{i=1}^{n} X_i(1-Y_i) \geqslant t \tag{4.11}$$

where X_i are independent and identically distributed (iid) according to the normal distribution $\mathcal{N}(\cdot;\mu,\sigma^2)$ and Y_i are also iid and follow the Bernoulli distribution $\mathrm{Ber}(1-p_{fc})$ with outcomes $\{0,1\}$. Each X_i can be regarded as a loss variable associated with a failure event with probability p_{fc}. The failure event is further defined as the total loss exceeding the predefined threshold t. Let $\widetilde{X}_i = -X_i \sim \mathcal{N}(\cdot;-\mu,\sigma^2)$, $\widetilde{t} = -t$, and $\widetilde{Y}_i = 1-Y_i \sim \mathrm{Ber}(p_{fc})$. The failure probability then becomes

$$\begin{aligned} p_f &= \mathbb{P}\left(\sum_{i=1}^{n} -X_i(1-Y_i) \leqslant -t\right) = \mathbb{P}\left(\sum_{i=1}^{n} \widetilde{X}_i\widetilde{Y}_i \leqslant \widetilde{t}\right) \\ &= \sum_{i=0}^{n} \mathbb{P}\left(\sum_{j=1}^{n} \widetilde{Y}_j = i\right) \mathbb{P}\left(\sum_{k=1}^{n} \widetilde{X}_k\widetilde{Y}_k \leqslant \widetilde{t} \,\middle|\, \sum_{j=1}^{n} \widetilde{Y}_j = i\right) \\ &= \sum_{i=1}^{n} \binom{n}{i} p_{fc}^{i}(1-p_{fc})^{n-i} \Phi\left(\frac{\widetilde{t}-(-\mu i)}{\sqrt{i\cdot\sigma^2}}\right) + (1-p_{fc})^n \mathbb{I}(\widetilde{t} \geqslant 0) \end{aligned} \tag{4.12}$$

Eq. (4.12) shows that the failure probability p_f is a function of n and $\widetilde{t}$ when fixing μ, σ and p_{fc}. For different n, we choose $\widetilde{t}$ such that the failure probability equals p_f^*. In this way, we define a series of failure events of different dimensions but with the same failure probability, p_f^*. Note that there is a 'jump' of value $(1-p_{fc})^n$ at the origin coordinate; this value decreases as the dimension increases. Fig. 4.5 shows the failure probability p_f as a function of $\widetilde{t}$ for the case $n = 1$ (the dimension is 2) and $n = 50$ (the dimension is 100). The failure probability can also be regarded as the CDF of $\sum_{i=1}^{n} \widetilde{X}_i\widetilde{Y}_i$.



(a) $n = 1$ (b) $n = 50$

Figure 4.5: Failure probability p_f vs. $\widetilde{t}$ for the flow problem of Example 5.2.

Next, MCS, standard SuS, and aE-SuS are carried out to obtain the failure probabilities. SuS and aE-SuS are performed in standard normal space with aCS as the MCMC algorithm. Here, we set $p_{fc} = p_f^* = 10^{-3}, \mu = -10, \sigma = 1$ and vary n from 1 to 50. tol, N_0 and p_0 for aE-SuS are set to be $0.5, 1{,}000, 0.1$, respectively. As shown in Fig. 4.6, the computational cost of aE-SuS, which is measured by the total number of LSF evaluations, decreases rapidly with increasing dimension and reaches around 4,000 calls of the LSF for higher dimensions. In order to obtain the statistical characteristics of the aE-SuS estimator and to compare them with MCS and standard SuS, 500 independent trials of aE-SuS and SuS are carried out. The results of MCS are calculated theoretically with the same computational cost (total number of samples N_{tot}) as aE-SuS. The MCS estimator is unbiased, and the coefficient of variation is $\frac{1-p_f}{\sqrt{N_{tot} \cdot p_f}}$. Fig. 4.7 and Fig. 4.8 illustrate the relative bias and the coefficient of variation of both aE-SuS and MCS. We see that the behavior of aE-SuS is similar to that of MCS in low dimensions where the jump at the origin is large, while in high dimensions where the jump of the CDF becomes smaller, aE-SuS is more efficient. We note that the influence of the number of random variables on the performance of the MCMC algorithm used in aE-SuS is insignificant. This is due to the fact that the aCS is specially designed for high-dimensional problems.

Fig. 4.9 compares the square root of MSE (RMSE, calculated through estimates of the two terms of Eq. (4.4)) of aE-SuS with different settings of standard SuS. It can be seen that even with well-tuned parameters ($p_0 = 1/40$), standard SuS can lead to significant errors in low to moderate dimensions where the jump in the CDF of the LSF is large.

As the 'jump' vanishes for large n, the results of aE-SuS become similar to that of standard SuS. In low dimensions, the aE-SuS algorithm behaves similarly to crude MCS.

4.5.3 Binomial experiment

This example studies the behavior of the independent M-H algorithm proposed in Section 3.2. Consider a binomial experiment with n trials. Each trial is an independent event that has two

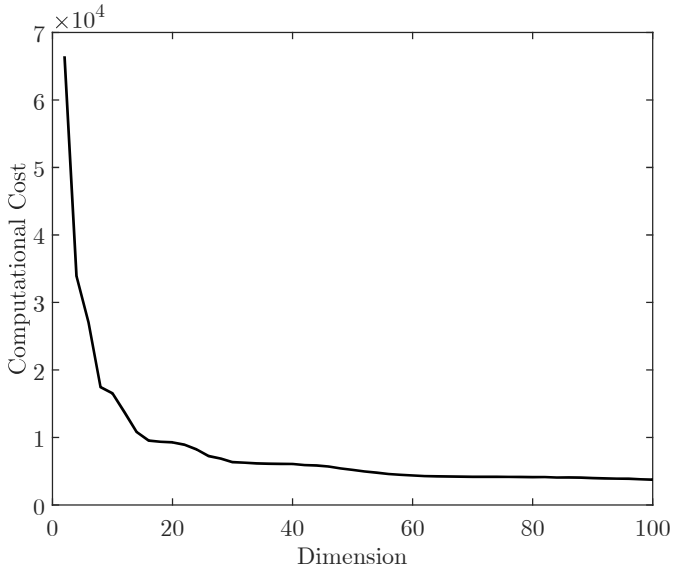


Figure 4.6: Computational cost of the aE-SuS for Example 5.2.

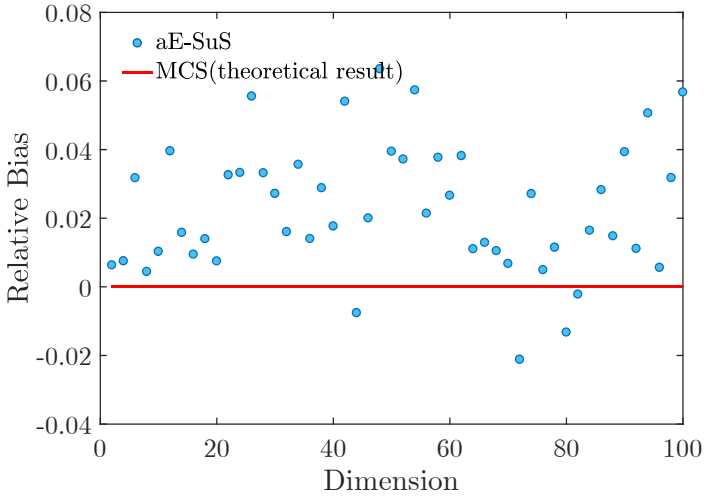


Figure 4.7: Relative bias (aE-SuS vs. MCS) for Example 5.2.

outcomes: 0 and 1. The probability that a trial is successful (takes outcome 1) is equal to p, and we evaluate the probability that at least t trials are successful. The LSF is then defined as:

$$g(\boldsymbol{X}) = t - \sum_{i=1}^{n} X_i$$

where X_i represents the outcome of the i-th trial. The exact failure probability can be expressed as $1 - F_B(t-1; n, p_{f_c})$ where $F_B(\cdot; n, p)$ is the CDF of the binomial distribution with parameters n and p.

In order to study the performance of the independent M-H algorithm in different dimensions, we fix p at 10^{-3} and vary n. For each n, aE-SuS with the independent M-H algorithm is run 200 times with parameters $tol = 0.8, p_0 = 0.1, N_0 = 2{,}000$.

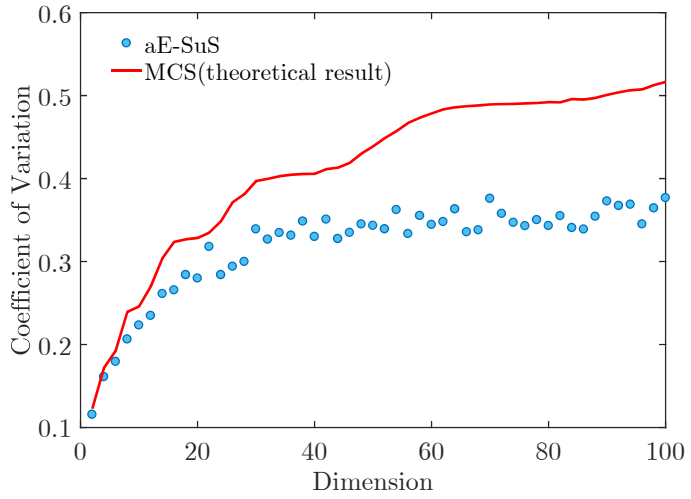


Figure 4.8: Coefficient of variation (aE-SuS vs. MCS) for Example 5.2.

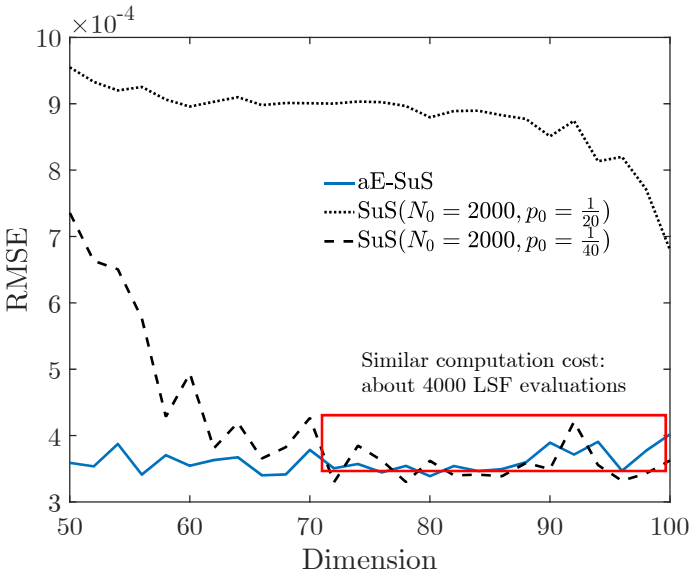


Figure 4.9: RMSE (aE-SuS vs. SuS) for Example 5.2.

For different dimensions and conditional levels of aE-SuS, the mean acceptance rate of the independent M-H algorithm, $\mathbb{E}(\alpha)$, is calculated through Eq. (4.10) and is summarized in Table 4.4. Note that $\mathbb{P}(\boldsymbol{X} \in F_l)$ and $\mathbb{P}(\boldsymbol{X} \notin \cup_{i=0,\ldots,l-1}\mathcal{X}_i)$ in Eq. (4.10) are abbreviated as $\mathbb{P}(F_l)$ and $\mathbb{P}(\overline{\cup\mathcal{X}_i})$, respectively. One can see that both $\mathbb{P}(F_l)$ and $\mathbb{P}(\overline{\cup\mathcal{X}_i})$ decrease as conditional level l increases. However, $\mathbb{P}(\overline{\cup\mathcal{X}_i})$ drops much slower than $\mathbb{P}(F_l)$ at high levels, which results in small $\mathbb{E}(\alpha)$. This is because, to form a good proposal in the independent M-H algorithm, the states that need to be excluded from the input distribution grow exponentially with l. This effect is more pronounced in high dimensions, leading to a faster decrease of $\mathbb{E}(\alpha)$. At the fourth level, the mean acceptance rate for $n = 100$ is only 0.2% of the rate for $n = 25$ in this example.

Nevertheless, if the mean acceptance rate is not too small, the independent M-H algorithm performs well. For instance, if we fix the threshold t at 4, the results of 200 independent runs of aE-SuS are summarized in Table 4.5. For all 4 cases, aE-SuS is slightly biased with less computational cost than

crude MCS for achieving the same coefficient of variation. Note that the average computational cost for $n = 25$ is much higher than the other three cases, which is due to the larger 'jumps' in the CDF of the LSF.

Table 4.4: Mean acceptance rate of the independent M-H algorithm for Example 5.3.

		$n = 25$	$n = 50$	$n = 75$	$n = 100$
$l = 1$ $(g(\boldsymbol{X}) < t)$	$\mathbb{P}(F_l)$	0.025	0.049	0.072	0.095
	$\mathbb{P}(\overline{\cup \mathcal{X}_i})$	0.025	0.049	0.072	0.095
	$\mathbb{E}(\alpha)$	1	1	1	1
$l = 2$ $(g(\boldsymbol{X}) < t - 1)$	$\mathbb{P}(F_l)$	$2.95 \cdot 10^{-4}$	0.0012	0.0026	0.0046
	$\mathbb{P}(\overline{\cup \mathcal{X}_i})$	$2.95 \cdot 10^{-4}$	0.0012	0.0026	0.0046
	$\mathbb{E}(\alpha)$	1	1	1	1
$l = 3$ $(g(\boldsymbol{X}) < t - 2)$	$\mathbb{P}(F_l)$	$2.26 \cdot 10^{-6}$	$1.89 \cdot 10^{-5}$	$6.40 \cdot 10^{-5}$	$1.50 \cdot 10^{-4}$
	$\mathbb{P}(\overline{\cup \mathcal{X}_i})$	$2.26 \cdot 10^{-6}$	$1.89 \cdot 10^{-5}$	$1.87 \cdot 10^{-4}$	0.0017
	$\mathbb{E}(\alpha)$	1	1	0.34	0.087
$l = 4$ $(g(\boldsymbol{X}) < t - 3)$	$\mathbb{P}(F_l)$	$1.24 \cdot 10^{-8}$	$2.22 \cdot 10^{-7}$	$1.15 \cdot 10^{-6}$	$3.63 \cdot 10^{-6}$
	$\mathbb{P}(\overline{\cup \mathcal{X}_i})$	$1.24 \cdot 10^{-8}$	$9.70 \cdot 10^{-6}$	$1.84 \cdot 10^{-4}$	0.0017
	$\mathbb{E}(\alpha)$	1	0.023	0.0063	0.0021

Table 4.5: Statistics of the aE-SuS estimator for Example 5.3.

	$n = 25$	$n = 50$	$n = 75$	$n = 100$
p_f	$1.24 \cdot 10^{-8}$	$2.22 \cdot 10^{-7}$	$1.15 \cdot 10^{-6}$	$3.63 \cdot 10^{-6}$
relative bias(%)	2	3	5	2
coefficient of variation	0.15	0.14	0.20	0.34
average cost	$8.23 \cdot 10^4$	$3.77 \cdot 10^4$	$2.68 \cdot 10^4$	$2.09 \cdot 10^4$
MCS cost	$3.58 \cdot 10^9$	$2.30 \cdot 10^8$	$2.17 \cdot 10^7$	$2.38 \cdot 10^6$

4.5.4 Power network system

In this example, we consider the IEEE39 bus benchmark system, which consists of 39 nodes and 46 weighted edges. The topology of the network is illustrated in Fig. 4.10 where orange circles represent the source nodes and grey circles represent the terminal nodes. Edges are weighted by their reactance values shown on the right-hand side of Fig. 4.10 and by their capacities shown on the left-hand side. The line capacity is modeled here as being proportional to the number of most efficient paths between any source and terminal node pair passing through that line. This example was previously investigated by Scherb et al. [36] to quantify the network reliability considering

cascading effects and spatially distributed hazards and by ro-Velasquez and Straub [33] to select representative failure scenarios.

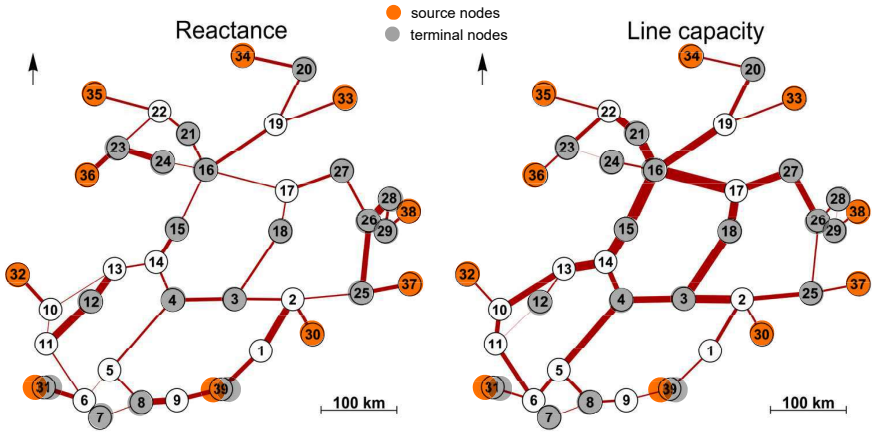


Figure 4.10: IEEE39 bus system, with edge thicknesses proportional to their estimated capacities (left) and reactances (right)[36].

The state of each node is considered as an independent Bernoulli random variable, with component failure probability randomly chosen from the uniform distribution $U[0, 10^{-2}]$. The LSF is then defined as a function of the system state $\boldsymbol{x}$, which is a binary vector, as follows:

$$g(\boldsymbol{x}) = \frac{E(\boldsymbol{x})}{E(\mathbf{1})} - \text{threshold} \tag{4.13}$$

$$E(\boldsymbol{x}) = \frac{1}{|SN||TN|} \sum_{s \in SN, t \in TN, t \neq s} eff_{st}(\boldsymbol{x}) \tag{4.14}$$

eff_{st} is the efficiency of the most efficient path from source node s to terminal node t, which is evaluated as the inverse of the sum of the reactance values along that path. $E(\boldsymbol{x})$ is the efficiency of the whole system associated with the system state $\boldsymbol{x}$ (The vector $\mathbf{1}$ is the intact system state). It is equal to the mean value of all the eff_{st} from each source node in set SN to each terminal node in set TN.

In order to model cascading effects, Eq. (4.13) is modified to

$$g(\boldsymbol{x}) = \frac{E(\mathcal{C}(\boldsymbol{x}))}{E(\mathbf{1})} - \text{threshold} \tag{4.15}$$

where $\mathcal{C}(\boldsymbol{x})$ is the final system state after cascading effects due to overloading of system components. These are triggered by overloading in individual lines following initial failures and are modeled following [36] and [15].

The threshold is fixed to 0.3, which means the system is considered as failed when its efficiency is less than 30% of that of the intact system. We apply the aE-SuS algorithm in the original Bernoulli space and set the parameters $N = 2{,}000, p_0 = 0.1, tol = 0.8$. The MCMC algorithm is the independent M-H algorithm. Fig. 4.11 shows the empirical CDF of $g(\boldsymbol{X})$ obtained by MCS and the aE-SuS algorithm, respectively. The aE-SuS algorithm is run 200 times to obtain the mean value, 10 percentile, and 90 percentile of the empirical CDF, while a single MCS run with 10^6 samples is carried out for validation.

The average computational cost of aE-SuS is 9,507 calculations of the LSF $g(\cdot)$, and the relative bias of the failure probability is 0.9%, while the coefficient of variation is 0.338. To achieve the same coefficient of variation as aE-SuS, crude MCS needs $1.74 \cdot 10^5$ calculations of the LSF in theory, which is significantly larger than that of aE-SuS. The average CPU time over 200 repetitions of aE-SuS is reported as 682 seconds on a 3.50GHz Intel Xeon E3-1270v3 computer. As a comparison, the CPU time for crude MCS with $1.74 \cdot 10^5$ samples on the same machine is $9.83 \cdot 10^3$ seconds, which is about 14 times larger than the CPU time of aE-SuS.

The standard SuS algorithm is not applicable for this example due to the large jump in the CDF of the LSF.

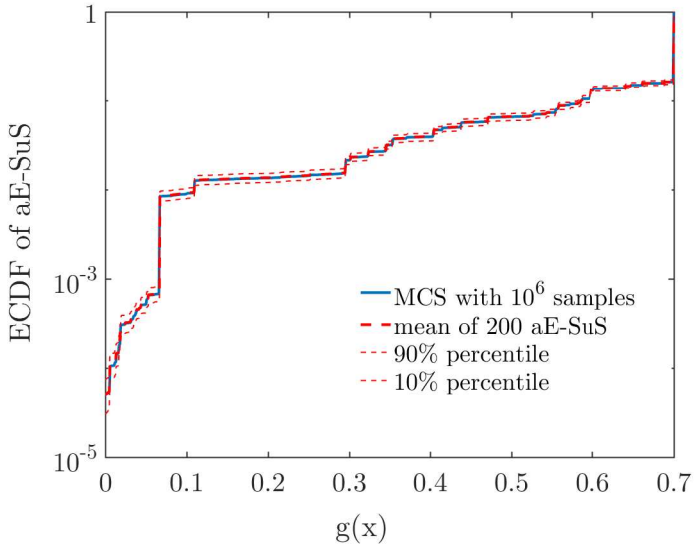


Figure 4.11: Results obtained by aE-SuS and MCS for IEEE39 network of Example 5.4.

4.6 Conclusions

We introduce adaptive effort subset simulation, which enables solving reliability problems with performance functions that follow a discontinuous distribution. Such problems often occur in network reliability assessment because of discrete random variables appearing in the input random vector or due to discontinuities in the function that defines the system performance. The proposed method modifies the adaptive selection of the intermediate domains of the standard SuS and adapts the number of samples and the respective conditional probability throughout the simulation to ensure that there is an adequate number of seeds at each level.

Any MCMC algorithm that enables efficient conditional sampling can be combined with the proposed algorithm. We implement the aCS algorithm in an underlying standard normal space. If the input random variables are all discrete, we propose a more efficient yet less general independent M-H algorithm, which operates in the original space. The mean acceptance rate of the independent M-H algorithm tends to decrease with an increase of the intermediate simulation level with a decreasing rate depending on the dimension of the input space. Hence, the algorithm becomes inefficient in

estimating small probabilities of high-dimensional systems. The acceptance rate of the aCS algorithm is independent of the input dimension. However, aCS performs worse than the independent M-H algorithm in moderate dimensional discrete input spaces.

Numerical results demonstrate that the aE-SuS estimator is only slightly biased and has substantially higher efficiency than crude Monte Carlo in problems where standard SuS fails to converge. The computational cost of the aE-SuS algorithm depends highly on the magnitude of the jumps in the distribution of the LSF.

4.7 Acknowledgment

The first author gratefully acknowledges the financial support of the China Scholarship Council.

4.A Adaptive conditional sampling algorithm

The aCS algorithm presented herein differs from the one of [28], which assumes that $N = \frac{1-p_0}{p_0} N_{seed}$. N is the number of new generated samples and N_{seed} represents the number of seeds. The implementation of the aCS is summarized in Algorithm 4. In the algorithm, l is the intermediate level, and n is the dimension. F_l is the intermediate event. $\mathcal{S}_l$ represents the seeds, and λ_{l-1} is the updated scaling parameter at the $l-1$-th level. It is suggested in [28] to choose λ_0 as 0.6. a^*, N_a, $\{\sigma_{0,d}\}_{d=1}^n$ are respectively the optimal acceptance rate, number of chains to consider for adaption, and the starting values for standard deviation. The suggested values can also be found in [28].

4.B Bound-based sampling algorithm

The original bound-based sampling algorithm [34, 26] is proposed for connectivity-based problems in multivariate Bernoulli spaces. However, it can be modified to sample from the proposal distribution at level l of the independent M-H algorithm of Section 3.2, $Q_p^{(l)}(\boldsymbol{x}) \propto Q(\boldsymbol{x}|F_0)\mathbb{I}\{\boldsymbol{x} \notin \cup_{j=0,\dots,l-1}\mathcal{X}_j\}$, if the input random variables follow the multivariate categorical distribution. That is

$$Q(\boldsymbol{x}|F_0) = \prod_{d=1}^{n}\sum_{i=1}^{n_d}\mathbb{I}\{x_d = i\}\theta_{d,i} \tag{16}$$

where $\theta_{d,i}$ represents the probability that the d-th component x_d equals value i given all preceding components $x_1, \dots, x_{d-1}$. n and n_d represent the number of components and the number of the states of x_d, respectively. For each component d, it holds that $\sum_{i=1}^{n_d}\theta_{d,i} = 1$.

The bound-based sampling algorithm proceeds in a component-wise scheme and is shown in Algorithm 5. Following this algorithm, one generates samples in the space $\{\boldsymbol{x} : \boldsymbol{x} \notin \cup_{j=0,\dots,l-1}\mathcal{X}_j\}$ with probability proportional to the input distribution $Q(\boldsymbol{x}|F_0)$. A detailed proof can be found in [26].

Algorithm 4: Adaptive conditional sampling algorithm

Input: N, F_l, $\mathcal{S}^{(l)}$, λ_{l-1}

$N_s \leftarrow |\mathcal{S}^{(l)}|, \lambda \leftarrow \lambda_{l-1}$

Define a^*, N_a, $\{\sigma_{0,d}\}_{d=1}^n$ according to [28]

Randomly sort the seeds $\mathcal{S}^{(l)}$

Randomly choose $\mathrm{mod}(N, N_s)$ elements from the set $\{1, 2, \cdots, N_s\}$, say $\boldsymbol{\chi}$

$\mathcal{P}^{(l)} \leftarrow \emptyset$, $c_1 \leftarrow 0$, $c_2 \leftarrow 0$

for $i = 1, \cdots, N_s$ **do**

  $\sigma_d \leftarrow \min(1, \lambda\sigma_{0,d})$, $\rho_d \leftarrow \sqrt{1-\sigma_d^2}$; $\quad d = 1, \cdots, n$

  if $i \in \boldsymbol{\chi}$ **then**

    $j \leftarrow \lfloor \frac{N}{N_s} \rfloor + 1$

  else

    $j \leftarrow \lfloor \frac{N}{N_s} \rfloor$

  $\boldsymbol{u}_0 \leftarrow$ the i-th seed

  for $k = 1, \cdots, j$ **do**

    for $d = 1, \cdots, n$ **do**

      %% sample the d-th component of candidate $\boldsymbol{v}$

      $v_d \leftarrow \mathcal{N}(\cdot; \rho_d u_{k-1,d}, \sigma_d^2)$

    if $\boldsymbol{v} \in F_l$ **then**

      $\boldsymbol{u}_k \leftarrow \boldsymbol{v}$

      $c_1 \leftarrow c_1 + 1$, $c_2 \leftarrow c_2 + 1$

    else

      $\boldsymbol{u}_k \leftarrow \boldsymbol{u}_{k-1}$

      $c_1 \leftarrow c_1 + 1$

    Add $\boldsymbol{u}_k$ to $\mathcal{P}^{(l)}$

  if *i is a multiple of N_a* **then**

    $\lambda \leftarrow \lambda \exp((\frac{i}{N_a})^{-1/2}[\frac{c_2}{c_1} - a^*])$

    $c_1 \leftarrow 0$, $c_2 \leftarrow 0$

$\lambda_l \leftarrow \lambda$

Output: $\mathcal{P}^{(l)}$, λ_l

Algorithm 5: Bound-based sampling algorithm

Input: $Q(\boldsymbol{x}|F_0)$

for $d = 1, \cdots, n$ **do**

  Calculate $\Pr(\boldsymbol{x} \notin \cup_{j=0,\ldots,l-1}\mathcal{X}_j | x_1, \cdots, x_{d-1})$

  for $i = 1, \cdots, n_d$ **do**

    Calculate $\Pr(\boldsymbol{x} \notin \cup_{j=0,\ldots,l-1}\mathcal{X}_j | x_1, \cdots, x_{d-1}, x_d = i)$

    $\theta^*_{d,i} \leftarrow \theta_{d,i} \frac{\Pr(\boldsymbol{x} \notin \cup_{j=0,\ldots,l-1}\mathcal{X}_j | x_1, \cdots, x_{d-1}, x_d = i)}{\Pr(\boldsymbol{x} \notin \cup_{j=0,\ldots,l-1}\mathcal{X}_j | x_1, \cdots, x_{d-1})}$

  Sample x_d from the categorical distribution $\mathrm{Cat}(x; \boldsymbol{\theta}^*_d)$

Output: candidate $\boldsymbol{x}$

References

[1] C. Alexopoulos and G. S. Fishman. "Characterizing stochastic flow networks using the Monte Carlo method". In: *Networks* 21.7 (1991), pp. 775–798.

[2] C. Andrieu and J. Thoms. "A tutorial on adaptive MCMC". In: *Statistics and Computing* 18.4 (2008), pp. 343–373.

[3] S.-K. Au. "On MCMC algorithm for subset simulation". In: *Probabilistic Engineering Mechanics* 43 (2016), pp. 117–120.

[4] S.-K. Au and J. L. Beck. "Estimation of small failure probabilities in high dimensions by subset simulation". In: *Probabilistic Engineering Mechanics* 16.4 (2001), pp. 263–277.

[5] M. O. Ball, C. J. Colbourn, and J. S. Provan. "Network reliability". In: *Handbooks in Operations Research and Management Science* 7 (1995), pp. 673–762.

[6] Z. I. Botev and D. P. Kroese. "An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting". In: *Methodology and Computing in Applied Probability* 10.4 (2008), pp. 471–505.

[7] Z. I. Botev and D. P. Kroese. "Efficient Monte Carlo simulation via the generalized splitting method". In: *Statistics and Computing* 22.1 (2012), pp. 1–16.

[8] Z. I. Botev, P. L'Ecuyer, G. Rubino, R. Simard, and B. Tuffin. "Static network reliability estimation via generalized splitting". In: *INFORMS Journal on Computing* 25.1 (2013), pp. 56–71.

[9] Z. I. Botev, P. L'Ecuyer, and B. Tuffin. "Reliability estimation for networks with minimal flow demand and random link capacities". In: *arXiv preprint arXiv:1805.03326* (2018).

[10] S. Bulteau and M. El Khadiri. "A new importance sampling Monte Carlo method for a flow network reliability problem". In: *Naval Research Logistics (NRL)* 49.2 (2002), pp. 204–228.

[11] J.-E. Byun and J. Song. "Generalized matrix-based Bayesian network for multi-state systems". In: *Reliability Engineering & System Safety* 211 (2021), p. 107468.

[12] H. Cancela and M. El Khadiri. "A recursive variance-reduction algorithm for estimating communication-network reliability". In: *IEEE Transactions on Reliability* 44.4 (1995), pp. 595–602.

[13] P.-C. Chang. "MC-based simulation approach for two-terminal multi-state network reliability evaluation without knowing d-MCs". In: *Reliability Engineering & System Safety* 220 (2022), p. 108289.

[14] J. Ching and W.-C. Hsu. "An efficient method for evaluating origin-destination connectivity reliability of real-world lifeline networks". In: *Computer-Aided Civil and Infrastructure Engineering* 22.8 (2007), pp. 584–596.

[15] P. Crucitti, V. Latora, and M. Marchiori. "Model for cascading failures in complex networks". In: *Physical Review E* 69.4 (2004), p. 045104.

[16] T. Elperin, I. Gertsbakh, and M. Lomonosov. "Estimation of network reliability using graph evolution models". In: *IEEE Transactions on Reliability* 40.5 (1991), pp. 572–581.

[17] G. S. Fishman. "A Monte Carlo sampling plan for estimating network reliability". In: *Operations Research* 34.4 (1986), pp. 581–594.

[18] G. S. Fishman. "Monte Carlo estimation of the maximal flow distribution with discrete stochastic arc capacity levels". In: *Naval Research Logistics (NRL)* 36.6 (1989), pp. 829–849.

[19] G. S. Fishman. "The distribution of maximum flow with applications to multistate reliability systems". In: *Operations Research* 35.4 (1986), pp. 607–618.

[20] G. S. Fishman and T.-Y. D. Shaw. "Evaluating reliability of stochastic flow networks". In: *Probability in the Engineering and Informational Sciences* 3.4 (1989), pp. 493–509.

[21] I. Gertsbakh, R. Rubinstein, Y. Shpungin, and R. Vaisman. "Permutational methods for performance analysis of stochastic flow networks". In: *Probability in the Engineering and Informational Sciences* 28.1 (2014), pp. 21–38.

[22] W. R. Gilks and P. Wild. "Adaptive rejection sampling for Gibbs sampling". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 41.2 (1992), pp. 337–348.

[23] G. Hardy, C. Lucet, and N. Limnios. "K-terminal network reliability measures with binary decision diagrams". In: *IEEE Transactions on Reliability* 56.3 (2007), pp. 506–515.

[24] J. He. "An extended recursive decomposition algorithm for dynamic seismic reliability evaluation of lifeline networks with dependent component failures". In: *Reliability Engineering & System Safety* 215 (2021), p. 107929.

[25] K.-P. Hui, N. Bean, M. Kraetzl, and D. P. Kroese. "The cross-entropy method for network reliability estimation". In: *Annals of Operations Research* 134.1 (2005), pp. 101–118.

[26] P. L'Ecuyer, G. Rubino, S. Saggadi, and B. Tuffin. "Approximate zero-variance importance sampling for static network reliability estimation". In: *IEEE Transactions on Reliability* 60.3 (2011), pp. 590–604.

[27] L. Murray, H. Cancela, and G. Rubino. "A splitting algorithm for network reliability estimation". In: *IIE Transactions* 45.2 (2013), pp. 177–189.

[28] I. Papaioannou, W. Betz, K. Zwirglmaier, and D. Straub. "MCMC algorithms for subset simulation". In: *Probabilistic Engineering Mechanics* 41 (2015), pp. 89–103.

[29] I. Papaioannou, S. Geyer, and D. Straub. "Improved cross-entropy-based importance sampling with a flexible mixture model". In: *Reliability Engineering & System Safety* 191 (2019), p. 106564.

[30] R. Paredes, L. Dueñas-Osorio, K. S. Meel, and M. Y. Vardi. "Principled network reliability approximation: A counting-based approach". In: *Reliability Engineering & System Safety* 191 (2019), p. 106472.

[31] J. E. Ramirez-Marquez and D. W. Coit. "A Monte Carlo simulation approach for approximating multi-state two-terminal reliability". In: *Reliability Engineering & System Safety* 87.2 (2005), pp. 253–264.

[32] M. Rausand and A. Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications*. John Wiley & Sons, 2003.

[33] H. Rosero-Velásquez and D. Straub. "Representative Natural Hazard Scenarios for Risk Assessment of Spatially Distributed Infrastructure Systems". In: *The 29th European Safety and Reliability Conference*. 2019, pp. 1–7.

[34] G. Rubino and B. Tuffin. *Rare Event Simulation Using Monte Carlo Methods*. John Wiley & Sons, 2009.

[35] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, 2016.

[36] A. Scherb, L. Garrè, and D. Straub. "Reliability and component importance in networks subject to spatially distributed hazards followed by cascading failures". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 3.2 (2017).

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

[38] Z. Wang, M. Broccardo, and J. Song. "Hamiltonian Monte Carlo methods for subset simulation in reliability analysis". In: *Structural Safety* 76 (2019), pp. 51–67.

[39] W.-C. Yeh. "A quick BAT for evaluating the reliability of binary-state networks". In: *Reliability Engineering & System Safety* 216 (2021), p. 107917.

[40] W. Yu, W. Huang, K. Wen, J. Zhang, H. Liu, K. Wang, J. Gong, and C. Qu. "Subset simulation-based reliability analysis of the corroding natural gas pipeline". In: *Reliability Engineering & System Safety* 213 (2021), p. 107661.

[41] Y. Zhou, L. Liu, and H. Li. "Reliability estimation and optimisation of multistate flow networks using a conditional Monte Carlo method". In: *Reliability Engineering & System Safety* 221 (2022), p. 108382.

[42] K. M. Zuev, S. Wu, and J. L. Beck. "General network reliability problem and its efficient solution by subset simulation". In: *Probabilistic Engineering Mechanics* 40 (2015), pp. 25–35.

Chapter 5

Bayesian improved cross entropy method for network reliability assessment

Original Publication

J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method for network reliability assessment". In: *Structural Safety* 103 (2023), p. 102344.

Abstract

We identify the zero count problem (or overfitting) of cross-entropy-based methods in the context of network reliability assessment and propose a consistent Bayesian estimator that mitigates this issue. Thereby, we derive the posterior predictive distribution of importance sampling distribution parameters that replaces the weighted maximum likelihood estimation employed in the standard cross entropy optimization. For rare event estimation, we embed the Bayesian estimator into the improved cross entropy (iCE) method and provide theoretical insights into the adaptive selection of intermediate target distributions in the iCE. The modified version of the iCE, termed the Bayesian iCE (BiCE), is proved to be unbiased. By contrast, even with higher computational costs, the standard iCE method is often significantly biased when solving network reliability problems. Our numerical investigations indicate that a uniform prior in the proposed BiCE method performs sub-optimally, and an informative symmetric Dirichlet prior is suggested.

5.1 Introduction

Infrastructure networks, such as power transmission networks and water supply systems, operate as the backbones of urban communities. Hence, it is essential to properly quantify the risk of network failure, which involves quantification of the failure probability of the network system. A network is considered as failed when it cannot deliver a specified level of performance. Mathematically, the failure is described through a function $g(\cdot)$, known as a performance function, structure function, or limit state function (LSF). Let $\boldsymbol{X}$ be a n-dimensional vector of random variables with joint density function $p_{\boldsymbol{X}}(\boldsymbol{x})$. The failure event F collects all system states $\boldsymbol{x}$ whose LSF $g(\boldsymbol{x})$ is less than or equal to 0, i.e., $F \triangleq \{\boldsymbol{x} : g(\boldsymbol{x}) \leq 0\}$. The probability of failure is defined as

$$p_f \triangleq \mathbb{P}(g(\boldsymbol{X}) \leq 0) = \mathbb{E}_p[\mathbb{I}\{g(\boldsymbol{X}) \leq 0\}], \tag{5.1}$$

where $\mathbb{I}\{\cdot\}$ represents the indicator function, and $\mathbb{E}_p$ denotes expectation with respect to the density $p_{\boldsymbol{X}}(\boldsymbol{x})$.

The network performance is often measured through connectivity or flow. In connectivity-based problems, one evaluates the probability that a given set of nodes are disconnected [34, 5], and typically, both the network performance and the component state are modeled as binary random variables. In flow-based problems, one is interested in the flow that a network can deliver, e.g., the maximum number of passengers that can be transported from one city to another through the railway network. Multi-state or continuous random variables are often involved in this class of problems. For water supply systems and power grids, the flow is driven by the physical law and operation strategies, and the initial failure of network components leads to a reconfiguration of the power flow that may trigger additional cascading failure.

A set of methodologies have been proposed for evaluating the reliability in the above two classes of problems, among which sampling-based methods such as Monte Carlo simulation (MCS) and its different variants feature prominently [31, 21, 22, 1, 19, 13, 44, 50, 6]. For rare event simulation, i.e., when the failure probability p_f is small, crude MCS is inefficient or even infeasible when the LSF is expensive to compute. In such cases, advanced sampling techniques such as subset simulation [51, 8, 52, 9, 28, 14] and importance sampling (IS) [11, 26, 49] should be employed to decrease the required number of LSF evaluations for obtaining an accurate estimate of p_f. An alternative is the use of surrogate-based methods to construct approximation models, also known as surrogate models, that imitate the behavior of the computationally demanding LSF. However, for network reliability problems, where the inputs are discrete, and the LSF is often discontinuous, the crucial assumptions of smoothness and regularity of classical surrogate models such as Kriging and polynomial chaos expansions are not always met [37]. Alternatively, Dehghani et. al. [18] employ Bayesian additive regression trees as the surrogate model for the reliability analysis of power grids. However, when the failure probability is small, e.g., of magnitude $10^{-4} \sim 10^{-7}$ as in our paper, their method requires a large initial sample set, which can introduce significant overhead.

In the present paper, we employ the IS technique for rare event estimation in static (or time independent) network reliability problems. In particular, we focus on an enhanced version of the multi-level cross entropy (CE) method [45], termed the improved cross entropy method (iCE) [40]. The basic idea of iCE is to perform a smooth transition from the input density to the optimal IS density via a parametric distribution model whose Kullback-Leibler (KL) divergence from the optimal IS is iteratively minimized. In the context of network reliability assessment with discrete multi-state

components, the obvious choice of the parametric family is the categorical distribution. However, updating the categorical model with the CE or iCE method can perform poorly, especially when the sample size is small. This is because the probability assigned to a certain category often converges to 0 when no samples fall into this category during the adaptive sampling process. This is known in the literature as the *zero count problem* [38]. Neglecting a certain category in the IS distribution can lead to a bias in the IS estimate of p_f. To avoid such an issue, one may think of transferring the discrete random variable space to a continuous one through, for example, the Rosenblatt transformation [14] and employ continuous parametric families in the iCE method. However, the network reliability problem becomes more challenging after this non-linear transformation, and the iCE method often fails to converge. Hui et. al. [27] combine the cross entropy method with the graph creation process [20] and efficiently estimate the connectivity reliability of networks using an independent exponential parametric model. Note that this method is computationally demanding and applies only to coherent binary systems. In this paper, we employ the independent categorical distribution as IS distribution and propose an approach for learning its parameters during the iCE sampling process that avoids the zero count problem. The proposed algorithm, termed Bayesian improve cross entropy method (BiCE), employs the posterior predictive distribution to update the parametric family instead of the weighted maximum likelihood estimator used in the standard CE method. Compared with other non-sampling-based methods (e.g., [33, 25, 41, 36, 12]), the proposed BiCE method facilitates using advanced network analysis algorithms that account for complex network dynamics. However, the BiCE may require a large number of samples to achieve acceptable results.

The rest of the paper is organized as follows. A brief introduction to the IS approach is given in Section 2. In Section 3, we review the CE and iCE methods and provide some new insights into these two methods. In Section 4, we first illustrate the problem that occurs when updating the categorical distribution using CE or iCE, and then propose the BiCE method to circumvent this problem. A set of numerical examples is given in Section 5 to illustrate the efficiency and accuracy of the proposed approach.

5.2 Importance sampling

Estimation of p_f in Eq. (5.1) using crude MCS is straightforward; one generates N samples from the joint density function $p_{\boldsymbol{X}}(\boldsymbol{x})$ and then takes the sample mean of the indicator function as the unbiased estimator of p_f. The coefficient of variation (c.o.v.) of the MCS estimate equals $\sqrt{\frac{1-p_f}{N \cdot p_f}}$; therefore, for small p_f the required number of samples for achieving an accurate result is large. For rare event estimation, acceleration techniques to speed up the occurrence of the failure events are necessary. IS is an efficient and widely utilized method for the simulation of rare events. The basic idea of IS is to sample from a proposal distribution, also known as IS distribution, under which the rare event is more likely to occur and to correct the resulting bias in the estimate by multiplying each sample in the IS estimator with an appropriate likelihood ratio L [44]. Specifically, let $p_{IS}(\boldsymbol{x})$ denote the IS density and $\{\boldsymbol{x}_k\}_{k=1}^N$ be the N samples generated from $p_{IS}(\boldsymbol{x})$. The IS estimator of the failure probability in Eq. (5.1) reads

$$\widehat{p}_f = \frac{1}{N} \sum_{k=1}^{N} \mathbb{I}\{g(\boldsymbol{x}_k) \leq 0\} \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_k)}{p_{IS}(\boldsymbol{x}_k)}, \tag{5.2}$$

where the likelihood ratio (or IS weight) $L(\boldsymbol{x}) \triangleq \frac{p_{\boldsymbol{X}}(\boldsymbol{x})}{p_{IS}(\boldsymbol{x})}$ can be interpreted as an adjustment factor that compensates for the fact that samples are generated from $p_{IS}(\boldsymbol{x})$ instead of $p_{\boldsymbol{X}}(\boldsymbol{x})$ [39]. The IS estimator in Eq. (5.2) is unbiased if the failure domain F is included in the sample space of $p_{IS}(\boldsymbol{x})$ [39]. The variance of the estimator mainly depends on the choice of the IS distribution. A proper choice of the IS distribution can lead to a significantly smaller variance than that of crude MCS. Indeed, the theoretical optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ that results in zero variance of the estimator is equal to the input distribution conditional on the occurrence of the failure event. That is

$$p^*_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{p_{\boldsymbol{X}}(\boldsymbol{x})\mathbb{I}\{g(\boldsymbol{x}) \leq 0\}}{p_f} = p_{\boldsymbol{X}}(\boldsymbol{x}|F). \tag{5.3}$$

Unfortunately, $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ cannot be directly used since its analytical expression relies on a prior knowledge of the sought failure probability p_f. Nevertheless, the optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ still provides guidance for selecting an appropriate IS distribution. A common approach is to perform an initial first/second order reliability method analysis [35] or employ a Markov chain simulation algorithm [3] to form a distribution that resembles $p^*_{\boldsymbol{X}}(\boldsymbol{x})$. Alternatively, one can approximate $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ in an adaptive manner through the application of the CE or iCE methods, which are discussed in detail in Section 3.

5.3 Cross entropy and improved cross entropy method

5.3.1 Cross entropy method

The CE method determines the IS distribution in the estimator in Eq. (5.2) through minimizing the KL divergence between the theoretical optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ and a predefined parametric family of distributions. The KL divergence, which is also known as relative entropy, is a measure of how one distribution differs from another. Specifically, let $h(\boldsymbol{x};\boldsymbol{v})$ denote a family of parametric distributions, where $\boldsymbol{v} \in \mathcal{V}$ is a parameter vector. The KL divergence between $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ and $h(\boldsymbol{x};\boldsymbol{v})$ is defined as [46]

$$\begin{aligned} D(p^*_{\boldsymbol{X}}, h) &= \mathbb{E}_{p^*_{\boldsymbol{X}}}\left[\ln\left(\frac{p^*_{\boldsymbol{X}}(\boldsymbol{X})}{h(\boldsymbol{X};\boldsymbol{v})}\right)\right] \\ &= \mathbb{E}_{p^*_{\boldsymbol{X}}}[\ln(p^*_{\boldsymbol{X}}(\boldsymbol{X}))] - \mathbb{E}_{p^*_{\boldsymbol{X}}}[\ln(h(\boldsymbol{X};\boldsymbol{v}))]. \end{aligned} \tag{5.4}$$

In order to obtain a precise IS estimator, the KL divergence $D(p^*_{\boldsymbol{X}}, h)$ needs to be small. In fact, one can prove that the c.o.v. of the IS estimator, $\delta(\widehat{P}_f)$ is lower-bounded by [4]

$$\delta(\widehat{P_f}) \geq \sqrt{\frac{\exp(D(p^*_{\boldsymbol{X}}, h)) - 1}{N}}. \tag{5.5}$$

According to Eq. (5.5), if we require that $\delta(\widehat{P_f}) \leq 0.1$, the KL divergence $D(p^*_{\boldsymbol{X}}, h)$ should be less or equal than $\ln(1 + 0.01N)$. Conversely, a large KL divergence leads to a high c.o.v. and hence an imprecise result.

The CE method determines the optimal parameter vector $\boldsymbol{v}^*$ through minimizing the KL divergence of Eq. (5.4), i.e., through solving

$$\boldsymbol{v}^* = \arg\min_{\boldsymbol{v}\in\mathcal{V}} D(p^*_{\boldsymbol{X}}, h). \tag{5.6}$$

Since the first term on the right hand side of Eq. (5.4) does not depend on $\boldsymbol{v}$, Eq. (5.6) is equivalent to

$$\boldsymbol{v}^* = \arg\min_{\boldsymbol{v}\in\mathcal{V}} -\mathbb{E}_{p^*_{\boldsymbol{X}}}[\ln(h(\boldsymbol{X};\boldsymbol{v}))]. \tag{5.7}$$

Typically, the optimization problem in Eq. (5.7) is convex and can be solved by the Lagrange multiplier method [10]. However, the objective function depends on the optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$, which is not known in closed form, and therefore Eq. (5.7) cannot be solved analytically. Instead, we estimate $\boldsymbol{v}^*$ through solving an alternative objective function, which is introduced in the following. Substituting $p^*_{\boldsymbol{X}}$ in Eq. (5.7) with the expression of Eq. (5.3), one gets

$$\boldsymbol{v}^* = \arg\max_{\boldsymbol{v}\in\mathcal{V}} \mathbb{E}_p[\mathbb{I}\{g(\boldsymbol{X})\leq 0\}\ln(h(\boldsymbol{X};\boldsymbol{v}))] \tag{5.8}$$

The expectation in Eq. (5.8) can be approximated through IS, which gives the importance sampling counterpart of the CE optimization problem. That is

$$\widehat{\boldsymbol{v}} = \arg\max_{\boldsymbol{v}\in\mathcal{V}} \frac{1}{N}\sum_{k=1}^{N} \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_k)\mathbb{I}\{g(\boldsymbol{x}_k)\leq 0\}}{p_{ref}(\boldsymbol{x}_k)} \ln(h(\boldsymbol{x}_k;\boldsymbol{v})), \qquad \boldsymbol{x}_k \sim p_{ref}(\cdot). \tag{5.9}$$

Here, $p_{ref}(\boldsymbol{x})$ is the IS distribution used to estimate the expectation in Eq. (5.8) and is termed the reference distribution in the CE method [46]. Similarly to the original CE optimization problem, the optimization problem in Eq. (5.9) can also be solved by the Lagrange multiplier method.

One should distinguish $h(\boldsymbol{x};\boldsymbol{v}^*)$ from $h(\boldsymbol{x};\widehat{\boldsymbol{v}})$ in the CE method [16]. $h(\boldsymbol{x};\boldsymbol{v}^*)$ represents the distribution that has the smallest KL divergence $D(p^*_{\boldsymbol{X}},h)$ among a set of distributions and hence is termed the sub-optimal IS distribution. $h(\boldsymbol{x};\widehat{\boldsymbol{v}})$ is the distribution we use as the IS distribution, i.e., the distribution resulting from the solution of the optimization problem of Eq. (5.9). We term it the chosen IS distribution for the rest of the paper. Note that, as long as the parametric family is fixed, the 'distance' between the optimal IS distribution and the sub-optimal IS distribution is also fixed. The objective of the CE method is finding a good estimator $\widehat{\boldsymbol{v}}$ that is close to the optimal but inaccessible CE parameter $\boldsymbol{v}^*$.

Remark 5.3.1. *In general, if $h(\boldsymbol{x};\boldsymbol{v})$ is a properly parameterized exponential family, $\widehat{\boldsymbol{v}}$ can be interpreted as the self-normalized IS estimator of $\boldsymbol{v}^*$. The accuracy of the self-normalized IS estimator is measured by the effective sample size (ESS). For more details, we refer to 5.A.*

Remark 5.3.2. *$\widehat{\boldsymbol{v}}$ can also be interpreted as a weighted maximum likelihood estimation (MLE) of $\boldsymbol{v}$ [23] and therefore may suffer from the same drawbacks as MLE (e.g., overfitting). To circumvent the overfitting issue of $\widehat{\boldsymbol{v}}$, we propose a novel Bayesian estimator $\widetilde{\boldsymbol{\mu}}$ for the CE method in Section 5.4. The proposed estimator converges to $\boldsymbol{v}^*$ as the sample size goes to infinity.*

5.3.2 Cross entropy method for rare events and improved cross entropy method

The efficiency and accuracy of the CE method depend on the choice of the reference distribution $p_{ref}(\boldsymbol{x})$ in Eq. (5.9). A potential choice for $p_{ref}(\boldsymbol{x})$ is the input distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$. However, for the case where $F=\{\boldsymbol{x}: g(\boldsymbol{x})\leq 0\}$ is a rare event, sampling directly from $p_{\boldsymbol{X}}(\boldsymbol{x})$ will lead to a large number of zero indicators in Eq. (5.9), and, hence, an inaccurate result.

In such a case, the reference distribution can be chosen adaptively. Let $p^{(t)}(\boldsymbol{x}), t = 1, ..., T$ denote a sequence of intermediate target distributions that gradually approaches the optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$. The CE optimization problem is then solved iteratively by finding a good approximation to each intermediate target distribution, resulting in a sequence of CE parameter vectors $\{\widehat{\boldsymbol{v}}^{(t)}, t = 1, ..., T\}$ and distributions $\{h(\boldsymbol{x}; \widehat{\boldsymbol{v}}^{(t)}), t = 1, ..., T\}$. The distribution one obtains in the t-th iteration, $h(\boldsymbol{x}; \widehat{\boldsymbol{v}}^{(t)})$, is used as the reference distribution $p_{ref}(\boldsymbol{x})$ for the CE procedure in iteration $t + 1$. For the first iteration, the input distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$ is used as the reference distribution. In this way, one takes $h(\boldsymbol{x}; \widehat{\boldsymbol{v}}^{(T-1)})$ as the reference distribution $p_{ref}(\boldsymbol{x})$ for Eq. (5.9), and $h(\boldsymbol{x}; \widehat{\boldsymbol{v}}^{(T)})$ as the final IS distribution. The goal is to make $\widehat{\boldsymbol{v}}^{(T)}$ a good estimator of $\boldsymbol{v}^*$. Typically, the intermediate target distributions $p^{(t)}(\boldsymbol{x})$ are not predefined but are chosen adaptively during the iterations. Depending on the way of adaptively selecting $p^{(t)}(\boldsymbol{x})$, one distinguishes the (multilevel) CE method and its improved version, the improved cross entropy (iCE) method.

For the CE method, the intermediate target distributions are defined as:

$$p^{(t)}(\boldsymbol{x}) \triangleq \frac{1}{Z^{(t)}} p_{\boldsymbol{X}}(\boldsymbol{x}) \mathbb{I}\{g(\boldsymbol{x}) \leq \gamma^{(t)}\}, t = 1, ..., T \tag{5.10}$$

where $\{\gamma^{(t)}, t = 1, ...T\}$ is a parameter vector that satisfies $\gamma^{(t)} \geq 0$, and $Z^{(t)}$ is a normalizing constant. The CE optimization problem for Eq. (5.10) reads

$$\boldsymbol{v}^{(t,*)} = \arg\max_{\boldsymbol{v} \in \mathcal{V}} \mathbb{E}_p[\mathbb{I}\{g(\boldsymbol{X}) \leq \gamma^{(t)}\} \ln(h(\boldsymbol{X}; \boldsymbol{v}))]. \tag{5.11}$$

The sample counterpart of the CE optimization problem for Eq. (5.11) reads as follows:

$$\widehat{\boldsymbol{v}}^{(t)} = \arg\max_{\boldsymbol{v} \in \mathcal{V}} \frac{1}{N} \sum_{k=1}^{N} \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_k) \mathbb{I}\{g(\boldsymbol{x}_k) \leq \gamma^{(t)}\}}{p_{ref}(\boldsymbol{x}_k)} \ln(h(\boldsymbol{x}_k; \boldsymbol{v})), \quad \boldsymbol{x}_k \sim p_{ref}(\cdot). \tag{5.12}$$

In the t-th iteration, the CE method proceeds through the following three steps: (1) Generate a set of samples $\mathcal{P}^{(t)} \triangleq \{\boldsymbol{x}_k, k = 1, ..., N\}$ from the reference distribution $p_{ref}(\boldsymbol{x}) = p_{\boldsymbol{X}}(\boldsymbol{x})$ in the first iteration and $p_{ref}(\boldsymbol{x}) = h(\boldsymbol{x}, \widehat{\boldsymbol{v}}^{(t-1)})$ thereafter. (2) Calculate the LSF value $g(\cdot)$ for each $\boldsymbol{x}_k$. Set $\gamma^{(t)}$ as the sample ρ-quantile of $\{g(\boldsymbol{x}_k), k = 1, ..., N\}$. ρ represents a hyperparameter of the CE method and is typically chosen between 0.01 and 0.1 [30]. (3) Solve the optimization problem of Eq. (5.12) with $\mathcal{P}^{(t)}$ to get a new parameter vector $\widehat{\boldsymbol{v}}^{(t)}$. The above three steps are iterated until for some iteration T, $\gamma^{(T)} \leq 0$. One then sets $\gamma^{(T)} = 0$ and carries out step (3) one last time to get $\widehat{\boldsymbol{v}}^{(T)}$.

In the iCE method, the intermediate target distributions are defined as:

$$p^{(t)}(\boldsymbol{x}) \triangleq \frac{1}{Z^{(t)}} p_{\boldsymbol{X}}(\boldsymbol{x}) \Phi\left(-\frac{g(\boldsymbol{x})}{\sigma^{(t)}}\right), t = 1, ..., T \tag{5.13}$$

where $\sigma^{(t)} > 0$ and Φ is the cumulative distribution function (CDF) of the standard normal distribution. Note that $\lim_{\sigma \to 0}(\Phi(-\frac{g(\boldsymbol{x})}{\sigma})) = \mathbb{I}\{g(\boldsymbol{x}) \leq 0\}$, meaning that for a decreasing sequence $\sigma^{(1)} > \cdots > \sigma^{(T)}$, the sequence of distributions gradually approaches the optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$. We note that alternative smooth approximations of the indicator function could be used instead of Φ to define the intermediate target distributions [48].

The CE optimization problem for Eq. (5.13) reads

$$\boldsymbol{v}^{(t,*)} = \arg\max_{\boldsymbol{v} \in \mathcal{V}} \mathbb{E}_p[\Phi(-g(\boldsymbol{X})/\sigma^{(t)}) \ln(h(\boldsymbol{X}; \boldsymbol{v}))]. \tag{5.14}$$

The sample counterpart of Eq. (5.14) can then be expressed as

$$\widehat{\boldsymbol{v}}^{(t)} = \arg\max_{\boldsymbol{v}\in\mathcal{V}} \frac{1}{N}\sum_{k=1}^{N} \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_k)\Phi(-g(\boldsymbol{x}_k)/\sigma^{(t)})}{p_{ref}(\boldsymbol{x}_k)} \ln(h(\boldsymbol{x}_k;\boldsymbol{v})), \quad \boldsymbol{x}_k \sim p_{ref}(\cdot). \tag{5.15}$$

According to 5.A, when $h(\boldsymbol{x};\boldsymbol{v})$ represents a properly parameterized exponential family, $\widehat{\boldsymbol{v}}^{(t)}$ is a self-normalized IS estimator of $\boldsymbol{v}^{(t,*)}$, independent of the choice of the intermediate target distributions. For the iCE method, the weight function of the self-normalized IS estimator of $\boldsymbol{v}^{(t,*)}$ equals

$$W(\boldsymbol{x};\sigma^{(t)}) = \frac{p_{\boldsymbol{X}}(\boldsymbol{x})\Phi(-g(\boldsymbol{x})/\sigma^{(t)})}{p_{ref}(\boldsymbol{x})}. \tag{5.16}$$

A common choice for measuring the accuracy of a self-normalized IS estimator is the ESS, whose approximate expression is given in Eq. (45). With predefined sample size N, ESS is only a function of the c.o.v. of the weight, $\delta\left(W(\boldsymbol{X};\sigma^{(t)})\right), \boldsymbol{X} \sim p_{ref}(\boldsymbol{x})$, which further depends on the reference distribution $p_{ref}(\boldsymbol{x})$ and the parameter $\sigma^{(t)}$.

In the t-th iteration of iCE, one fixes the reference distribution $p_{ref}(\boldsymbol{x})$ as $h(\boldsymbol{x};\widehat{\boldsymbol{v}}^{(t-1)})$ (as $p_{\boldsymbol{X}}(\boldsymbol{x})$ in the first iteration) and selects $\sigma^{(t)}$ such that the sample c.o.v. of the weights $\{W(\boldsymbol{x}_k;\sigma^{(t)})\}_{k=1}^N$ equals a predefined target value δ_{tar}, i.e., one solves the following optimization problem:

$$\sigma^{(t)} = \arg\min_{\sigma\in(0,\sigma^{(t-1)})} |\widehat{\delta}\left(\{W(\boldsymbol{x}_k;\sigma)\}_{k=1}^N\right) - \delta_{tar}|, \qquad \boldsymbol{x}_k \sim p_{ref}(\boldsymbol{x}). \tag{5.17}$$

where $W(\boldsymbol{x}_k;\sigma)$ represents the weight in Eq. (5.16) and is a function of the optimization variable σ. In this way, the sample ESS equals $\frac{N}{1+\delta_{tar}^2}$. Hence, the accuracy of the self-normalized IS estimator $\widehat{\boldsymbol{v}}^{(t)}$ is tuned by the hyperparameter δ_{tar}. A large δ_{tar} leads to an inaccurate $\widehat{v}^{(t)}$, while a small δ_{tar} increases the number of the intermediate target distributions $p^{(t)}(\boldsymbol{x})$ required to approach the optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$, thereby reducing the overall efficiency of the iCE method. This will be illustrated in detail in Section 5.4. In general, 1.5 is a good choice for δ_{tar} in the iCE method [40]. Once $\sigma^{(t)}$ is fixed, the optimization problem of Eq. (5.15) can be solved for the parameter vector $\widehat{\boldsymbol{v}}^{(t)}$. The corresponding distribution $h(\boldsymbol{x};\widehat{\boldsymbol{v}}^{(t)})$ is then used as the reference distribution for the $(t+1)$-th iteration.

The above procedure is iterated until the c.o.v. of the likelihood ratio [39] for sampling from $p^{(t)}(\boldsymbol{x})$ instead of $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ is smaller than δ_ϵ, i.e.,

$$\delta\left(\frac{p^*_{\boldsymbol{X}}(\boldsymbol{X})}{p^{(t)}(\boldsymbol{X})}\right) = \delta\left(\frac{\mathbb{I}\{g(\boldsymbol{X})\le 0\}}{\Phi(-g(\boldsymbol{X})/\sigma^{(t)})}\right) \le \delta_\epsilon, \qquad \boldsymbol{X} \sim p^{(t)}(\boldsymbol{x}). \tag{5.18}$$

In practice, we sample $\mathcal{P}^{(t)} = \{\boldsymbol{x}_k\}_{k=1}^N$ from $h(\boldsymbol{x};\widehat{\boldsymbol{v}}^{(t)})$ rather than $p^{(t)}(\boldsymbol{x})$, and check whether the sample c.o.v. of $\frac{\mathbb{I}\{g(\boldsymbol{x}_k)\le 0\}}{\Phi(-g(\boldsymbol{x}_k)/\sigma^{(t)})}$ is less or equal than δ_ϵ. Typically, δ_ϵ is chosen the same as δ_{tar} [40].

The algorithm for the iCE method is shown in Algorithm 6.

Algorithm 6: Improved cross entropy algorithm

Input: N, δ_{tar}, δ_ϵ, LSF $g(\boldsymbol{x})$, input distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$

$t \leftarrow 1$, $t_{max} \leftarrow 50$, $\sigma_0 \leftarrow \infty$

$h(\boldsymbol{x}; \widehat{\boldsymbol{v}}^{(t-1)}) \leftarrow p_{\boldsymbol{X}}(\boldsymbol{x})$

while *true* **do**

  Generate N samples $\{\boldsymbol{x}_k\}_{k=1}^N$ from $h(\boldsymbol{x}; \widehat{\boldsymbol{v}}^{(t-1)})$ and calculate the corresponding LSF values $\{g(\boldsymbol{x}_k)\}_{k=1}^N$

  Compute the sample c.o.v. $\widehat{\delta}$ of $\left\{\frac{\mathbb{I}\{g(\boldsymbol{x}_k)\leq 0\}}{\Phi(-g(\boldsymbol{x}_k)/\sigma^{(t-1)})}\right\}_{k=1}^N$

  if $t > t_{max}$ *or* $\widehat{\delta} \leq \delta_\epsilon$ **then**

    Break

  Determine $\sigma^{(t)}$ through solving Eq. (5.17)

  Compute $\widehat{\boldsymbol{v}}^{(t)}$ through solving Eq. (5.15)

  $t \leftarrow t + 1$

$T \leftarrow t - 1$

Use $h(\boldsymbol{x}; \widehat{\boldsymbol{v}}^{(T)})$ as the IS distribution and calculate the IS estimator $\widehat{p}_f$ through Eq. (5.2)

Output: $\widehat{p}_f$

5.4 Bayesian improved cross entropy method for the categorical parametric family

In this section, we consider the iCE method for estimating a rare event with a discrete random input $\boldsymbol{X}$, which often occurs in network reliability assessment. For discrete inputs $\boldsymbol{X}$, the probability mass function of $\boldsymbol{X}$, $p_{\boldsymbol{X}}(\boldsymbol{x})$, defines the probability of the corresponding outcome, i.e., $p_{\boldsymbol{X}}(\boldsymbol{x}) = \Pr(\boldsymbol{X} = \boldsymbol{x})$. We consider a slightly different definition of the intermediate target distribution in Eq. (5.13), which results from the definition of an auxiliary LSF $g_a(\boldsymbol{x})$:

$$g_a(\boldsymbol{x}) \triangleq \begin{cases} g(\boldsymbol{x}), & \text{if} \quad g(\boldsymbol{x}) > 0 \\ 0, & \text{if} \quad g(\boldsymbol{x}) \leq 0 \end{cases}. \tag{5.19}$$

Note that the failure probability p_f is unchanged if the original LSF in Eq. (5.1) is substituted with the auxiliary one, so we can equivalently estimate the probability that $g_a(\boldsymbol{X}) \leq 0$ for p_f, i.e.,

$$p_f = \mathbb{P}(g_a(\boldsymbol{X}) \leq 0) = \sum_{\boldsymbol{x} \in \Omega_{\boldsymbol{X}}} p_{\boldsymbol{X}}(\boldsymbol{x}) \mathbb{I}\{g_a(\boldsymbol{x}) \leq 0\}, \tag{5.20}$$

where $\Omega_{\boldsymbol{X}}$ is the sample space of the input random variables. In this way, the intermediate target distribution in Eq. (5.13) becomes

$$p^{(t)}(\boldsymbol{x}) \triangleq \frac{1}{Z^{(t)}} p_{\boldsymbol{X}}(\boldsymbol{x}) \Phi\left(-\frac{g_a(\boldsymbol{x})}{\sigma^{(t)}}\right), t = 1, ..., T. \tag{5.21}$$

In the following, we discuss the properties of the iCE method with the intermediate target distribution in Eq. (5.21). In particular, we examine the adaptation of the intermediate target distribution following Eq. (5.17) and formulate a theorem stating that, under certain assumptions, the resulting distribution sequence gradually approaches the optimal IS distribution. For the independent

categorical parametric family, i.e., the joint distribution consisting of independent components that follow the categorical distribution, we further illustrate the overfitting issue of the standard iCE method. Based on this observation, we introduce a novel approach called the Bayesian improved cross entropy (BiCE) method that circumvents this problem.

5.4.1 Adaptation of the intermediate target distribution

The adaptation of the intermediate target distribution $p^{(t)}(\boldsymbol{x})$, or equivalently the parameter $\sigma^{(t)}$, plays an important role in achieving a balance between the efficiency and accuracy of the iCE method. Therefore, it is worth taking a closer look at the updating formula of $\sigma^{(t)}$ in Eq. (5.17) and (5.18). To simplify the problem, we make the following assumptions:

Assumption 1. *The intermediate target distributions $p^{(t)}(\boldsymbol{x}), t = 1, ..., T$ are included in the parametric family $h(\boldsymbol{x}; \boldsymbol{v})$ and therefore can be perfectly matched by $h(\boldsymbol{x}; \boldsymbol{v}^{(t,*)}), t = 1, ..., T$.*

Assumption 2. *The sample size is infinite such that $\widehat{\boldsymbol{v}}^{(t)}$ is the same as $\boldsymbol{v}^{(t,*)}$.*

Under these two assumptions, the sample c.o.v. of the weight in Eq. (5.17) converges to the true c.o.v. $\delta\left(\frac{\Phi(-g_a(\boldsymbol{X})/\sigma)}{\Phi(-g_a(\boldsymbol{X})/\sigma^{(t-1)})}\right), \boldsymbol{X} \sim p^{(t-1)}(\boldsymbol{x})$, which is a function of σ. We write this function as $\delta^{(t)}(\sigma)$ for the rest of this paper. According to Eq. (5.18), the adaptive procedure of the iCE method is stopped when $\delta^{(t)}(0) \leq \delta_\epsilon$. In [15], we introduce the following two theorems:

Theorem 5.4.1. *Under Assumptions 1 and 2, $\delta^{(t)}(\sigma)$ is a strictly decreasing function of σ over $[0, \sigma^{(t-1)}]$.*

Theorem 5.4.2. *Under Assumptions 1 and 2, it holds $\delta^{(t)}(\sigma^{(t-1)}) = 0$ and $\delta^{(t)}(0) = \sqrt{\frac{Z^{(t-1)}}{0.5p_f} - 1} > 0$, where $Z^{(t-1)}$ is the normalizing constant of $p^{(t-1)}(\boldsymbol{x})$ and can be expressed as*

$$Z^{(t-1)} = 0.5p_f + \sum_{g_a(\boldsymbol{x})>0} p_{\boldsymbol{X}}(\boldsymbol{x})\Phi\left(-\frac{g_a(\boldsymbol{x})}{\sigma^{(t-1)}}\right). \tag{5.22}$$

As a corollary, the optimization problem of Eq. (5.17) has a unique solution $\sigma^{(t)}$ that is smaller than $\sigma^{(t-1)}$, resulting in a strictly decreasing sequence of σ, i.e., $\sigma^{(1)} > \sigma^{(2)} > \cdots > \sigma^{(t)}$. Additionally, since $Z^{(t)}$ is a strictly increasing function of $\sigma^{(t)}$ according to Eq. (5.22), we have $Z^{(1)} > Z^{(2)} > \cdots > Z^{(t)}$, and this further leads to another decreasing sequence of $\delta^{(t)}(0)$, i.e., $\delta^{(1)}(0) > \delta^{(2)}(0) > \cdots > \delta^{(t)}(0)$, which is sure to converge but does not necessarily converge to zero. If for some iteration T it holds $\delta^{(T)}(0) \leq \delta_\epsilon$, we terminate the adaptive procedure of iCE. The adaptation of $\sigma^{(t)}$ under Assumptions 1 and 2 is intuitively illustrated in Fig. 5.1. All symbols in the figure have the same meaning as before.

In practice, the parametric family $h(\boldsymbol{x}; \boldsymbol{v})$ has limited flexibility, and the sample size is also finite due to limited computational budgets. Nevertheless, we expect that the results given in Theorems 4.1 and 4.2 still apply when $h(\boldsymbol{x}; \boldsymbol{v}^{(t)})$ forms a good approximation of $p^{(t)}(\boldsymbol{x})$, which is confirmed in the numerical experiments in Section 5.5. The adaptation of $p^{(t)}(\boldsymbol{x})$ in iCE is tuned by hyper-parameters δ_{tar} and δ_ϵ. A small δ_ϵ indicates a strict (even infeasible) convergence criterion for $p^{(t)}(\boldsymbol{x})$, which

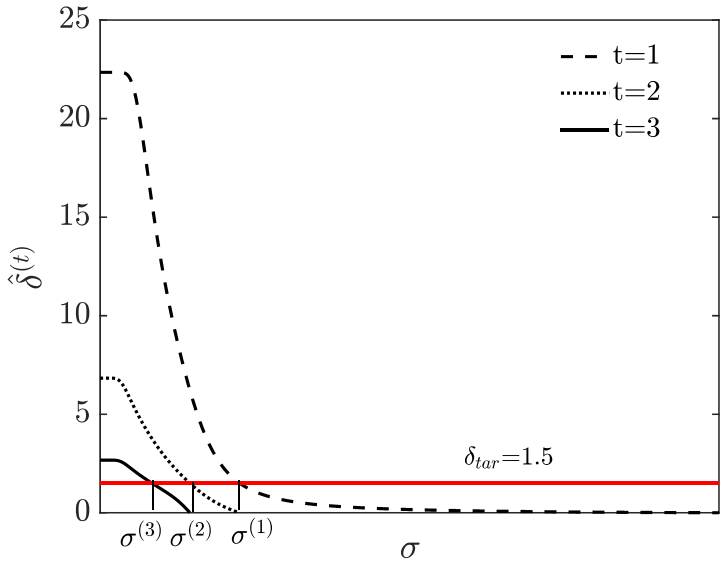


Figure 5.1: A schematic diagram of adaptive selection of $\sigma^{(t)}$.

leads to a more accurate yet less efficient result. Similarly, a small δ_{tar} leads to a $\sigma^{(t)}$ close to $\sigma^{(t-1)}$, which lowers the speed of the intermediate target distribution approaching the optimal IS distribution, thereby reducing the overall efficiency of the iCE algorithm. Conversely, a small δ_{tar} implies a large ESS and hence high accuracy of $\widehat{\boldsymbol{v}}^{(t)}$ in each t-th iteration of iCE method. In this paper, we suggest selecting $\delta_{tar} = \delta_\epsilon$ from 1 to 2 [40], which is justified by the numerical examples in Sec. 5.5. We also find that, instead of the original weight function defined in Eq. (5.16), using the alternative weight function

$$W^{alt}(\boldsymbol{x};\sigma) \triangleq \frac{\Phi(-g_a(\boldsymbol{x})/\sigma)}{\Phi(-g_a(\boldsymbol{x})/\sigma^{(t-1)})} \tag{5.23}$$

when solving $\sigma^{(t)}$ through Eq. (5.17) will lead to a better convergence of the iCE algorithm for network reliability assessment, especially when δ_ϵ is small.

5.4.2 Parametric distribution family for discrete inputs and zero count problem

To form a good approximation of $p^{(t)}(\boldsymbol{x})$ in the iCE method, a proper choice of the parametric family is necessary. In the context of the reliability of systems with multi-state components, the obvious choice of the parametric model is the multivariate categorical distribution, which assigns a probability to each system state of the network, i.e., to each possible state in the sample space of the input distribution. The multivariate categorical distribution has great flexibility as it includes all possible distributions defined in the sample space of the network components. However, the number of parameters of this model grows exponentially with the input dimension (number of components), making this model impractical even for moderate dimensions. Therefore, we consider independent categorical distributions.

Suppose $\boldsymbol{X}$ is a n dimensional input random vector with statistically independent components and each d-th component X_d follows the categorical distribution taking values $\{s_{d,1}, \cdots s_{d,n_d}\}$ with

probabilities $\{p_{d,1}, \cdots, p_{d,n_d}\}$. n_d is the number of sample states of X_d. The independent categorical family for $\boldsymbol{X}$ has the following general form:

$$h(\boldsymbol{x};\boldsymbol{v}) = \prod_{d=1}^{n} h_d(x_d;\boldsymbol{v}_d) = \prod_{d=1}^{n}\prod_{i=1}^{n_d} v_{d,i}^{\mathbb{I}\{x_d=s_{d,i}\}}, 0 \le v_{d,i} \le 1, \sum_{i=1}^{n_d} v_{d,i} = 1, \tag{5.24}$$

where $h_d(x_d;\boldsymbol{v}_d) = \prod_{i=1}^{n_d} v_{d,i}^{\mathbb{I}\{x_d=s_{d,i}\}}$ represents a univariate categorical distribution for X_d that assigns a probability of $v_{d,i}$ to each i-th state $s_{d,i}$ of X_d, and $\boldsymbol{v} = \{\boldsymbol{v}_d\}_{d=1}^{n}$ gathers the parameters of all components of the independent categorical family $h(\boldsymbol{x};\boldsymbol{v})$.

The optimal parameter $\boldsymbol{v}^{(t,*)}$ is obtained through solving Eq. (5.14), which gives[46]

$$v_{d,i}^{(t,*)} = \mathbb{E}_{p^{(t)}}[\mathbb{I}\{X_d = s_{d,i}\}]. \tag{5.25}$$

The explicit expression of $\boldsymbol{v}^{(t,*)}$ requires knowledge of the normalizing constant of $p^{(t)}$ and hence cannot be directly used in the iCE method. Through optimizing the alternative objective function in Eq. (5.15), the near-optimal parameter $\widehat{\boldsymbol{v}}^{(t)}$ is explicitly given by

$$\widehat{v}_{d,i}^{(t)} = \frac{\sum_{k=1}^{N} W(\boldsymbol{x}_k;\sigma^{(t)})\mathbb{I}\{x_{k,d} = s_{d,i}\}}{\sum_{k=1}^{N} W(\boldsymbol{x}_k;\sigma^{(t)})}, \quad d = 1, ..., n, \quad i = 1, ..., n_d, \tag{5.26}$$

where samples $\{\boldsymbol{x}_k\}_{k=1}^{N}$ are generated from the reference distribution $p_{ref}(\boldsymbol{x}) = h(\boldsymbol{x};\widehat{\boldsymbol{v}}^{(t-1)})$, and $W(\boldsymbol{x}_k;\sigma^{(t)}) = \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_k)\Phi(-g_a(\boldsymbol{x}_k)/\sigma^{(t)})}{p_{ref}(\boldsymbol{x}_k)}$. The expression of Eqs.(5.25) and (5.26) can also be obtained by considering that the independent categorical distribution is a member of the exponential family [38]. Note that $\widehat{\boldsymbol{v}}^{(t)}$ is the self-normalized estimator of $\boldsymbol{v}^{(t,*)}$. Additionally, $\widehat{\boldsymbol{v}}^{(t)}$ can be regarded as the weighted MLE of $\boldsymbol{v}$. This is because the objective function in Eq. (5.15) can be interpreted as a weighted log-likelihood function $\mathcal{LL}(\boldsymbol{v})$, with data set $\{\boldsymbol{x}_k\}_{k=1}^{N}$ and weights $\{W(\boldsymbol{x}_k)/N\}_{k=1}^{N}$.

Similarly to the MLE of independent categorical distribution, $\widehat{\boldsymbol{v}}^{(t)}$ suffers from overfitting, which is also known as the zero count problem in the context of MLE with categorical data [38], and results in poor performance when the sample size is small. In particular, if there is no sample whose d-th component equals $s_{d,i}$, $s_{d,i}$ will be assigned a zero probability according to Eq. (5.26), i.e., $\hat{v}_{d,i}^{(t)} = 0$. In the context of the iCE method, the parameter vector $\widehat{\boldsymbol{v}}^{(t)}$ is employed to generate samples at the $(t+1)$ iteration, and hence, $s_{d,i}$ will not occur in any of the newly generated samples. In this way, we have $\hat{v}_{d,i}^{(t)} = \hat{v}_{d,i}^{(t+1)} = \cdots = \hat{v}_{d,i}^{(T)} = 0$, resulting in a reduced sample space of the final IS distribution $h(\boldsymbol{x};\widehat{\boldsymbol{v}}^{(T)})$. However, for the optimal IS distribution, state $s_{d,i}$ is not necessarily negligible. In other words, the reduced sample space may only cover a part of the failure domain F, thereby underestimating the failure probability p_f.

5.4.3 Bayesian improved cross entropy method

In this subsection, we propose an accurate yet efficient algorithm termed the Bayesian improved cross entropy method (BiCE) that circumvents the zero count problem. In this approach, instead of employing a weighted MLE $\widehat{\boldsymbol{v}}^{(t)}$, a prior distribution is imposed on $\boldsymbol{v}^{(t)}$, and the posterior predictive distribution is derived, which is then employed to update the independent categorical family in iCE.

We insert the expression of the independent categorical parametric family of Eq. (5.24) into Eq. (5.15) and rewrite the objective function, or the weighted log-likelihood function $\mathcal{LL}(\boldsymbol{v})$, as follows

$$\begin{aligned}\mathcal{LL}(\boldsymbol{v}) &= \sum_{k=1}^{N} \frac{W(\boldsymbol{x}_k;\sigma^{(t)})}{N} \ln\left(\prod_{d=1}^{n} h_d(x_{k,d};\boldsymbol{v}_d)\right) \\ &= \sum_{d=1}^{n}\sum_{k=1}^{N} \frac{W(\boldsymbol{x}_k;\sigma^{(t)})}{N} \ln\left(h_d(x_{k,d};\boldsymbol{v}_d)\right) \\ &\triangleq \sum_{d=1}^{n} \mathcal{LL}_d(\boldsymbol{v}_d), \end{aligned} \tag{5.27}$$

where $\mathcal{LL}_d(\boldsymbol{v}_d)$ is the weighted log-likelihood function of a one-dimensional categorical family $h_d(x_d;\boldsymbol{v}_d)$, with data set $\{x_{k,d}\}_{k=1}^N$ and weights $\{\frac{W(\boldsymbol{x}_k;\sigma^{(t)})}{N}\}_{k=1}^N$. From Eq. (5.27), we find that, once the sample set is fixed, the parameter vectors $\boldsymbol{v}_d, d = 1,\cdots,n$, are decoupled from each other in the expression of $\mathcal{LL}(\boldsymbol{v})$, that is, the influence of each $\boldsymbol{v}_d$ on the outcome of $\mathcal{LL}(\boldsymbol{v})$ is separated (or additive). Additionally, we note that the feasible region for each parameter vector $\boldsymbol{v}_d$, that is $\mathcal{V}_d : \{0 \le v_{d,i} \le 1; i = 1,\cdots,n_d | \sum_{i=1}^{n_d} v_{d,i} = 1\}$, is independent. Therefore, the original optimization problem can be decomposed into n simpler subproblems, in which $\mathcal{LL}_d(\boldsymbol{v}_d)$ is maximized with respective to $\boldsymbol{v}_d \in \mathcal{V}_d$. The solutions to the subproblems are then concatenated to give a solution to the original problem, i.e., $\widehat{\boldsymbol{v}}^{(t)} = [\widehat{\boldsymbol{v}}_1^{(t)};\cdots;\widehat{\boldsymbol{v}}_n^{(t)}]$. Therefore, it is sufficient to discuss the following subproblem:

$$\widehat{\boldsymbol{v}}_d^{(t)} = \arg\max_{\boldsymbol{v}_d\in\mathcal{V}_d} \sum_{k=1}^{N} \frac{W(\boldsymbol{x}_k;\sigma^{(t)})}{N} \ln\left(h_d(x_{k,d};\boldsymbol{v}_d)\right).$$

Note that multiplying the objective function with a positive constant α or taking an exponential of the objective function does not change the solution to the optimization problem, so we have

$$\begin{aligned}\widehat{\boldsymbol{v}}_d^{(t)} &= \arg\max_{\boldsymbol{v}_d\in\mathcal{V}_d} \sum_{k=1}^{N} \frac{\alpha W(\boldsymbol{x}_k;\sigma^{(t)})}{N} \ln\left(h_d(x_{k,d};\boldsymbol{v}_d)\right) \\ &= \arg\max_{\boldsymbol{v}_d\in\mathcal{V}_d} \prod_{k=1}^{N} \left(h_d(x_{k,d};\boldsymbol{v}_d)\right)^{\frac{\alpha W(\boldsymbol{x}_k;\sigma^{(t)})}{N}}. \end{aligned} \tag{5.28}$$

The objective function in Eq. (5.28) can be regarded as a weighted likelihood function for $h_d(x_d;\boldsymbol{v}_d)$ with data set $\{x_{k,d}\}_{k=1}^N$ and weights $\{\frac{\alpha W(\boldsymbol{x}_k;\sigma^{(t)})}{N}\}_{k=1}^N$. In this work, α is chosen such that the weighted likelihood function coincides with the standard likelihood function with unit weights when $W(\boldsymbol{x}_k), k = 1,...,N$ are all equal, which gives

$$\alpha = \frac{N^2}{\sum_{k=1}^{N} W(\boldsymbol{x}_k;\sigma^{(t)})}. \tag{5.29}$$

Inserting the above expression of α and substituting the parametric family $h_d(x_{k,d};\boldsymbol{v}_d)$ with the expression in Eq. (5.24) into Eq. (5.28) gives

$$\widehat{\boldsymbol{v}}_d^{(t)} = \arg\max_{\boldsymbol{v}_d\in\mathcal{V}_d} \prod_{i=1}^{n_d} v_{d,i}^{\sum_{k=1}^{N}\left(\mathbb{I}\{x_{k,d}=s_{d,i}\}w_k\right)} \triangleq \arg\max_{\boldsymbol{v}_d\in\mathcal{V}_d} \mathcal{L}_d(\boldsymbol{v}_d), \tag{5.30}$$

where $w_k \triangleq N\frac{W(\boldsymbol{x}_k;\boldsymbol{\sigma}^{(t)})}{\sum_{k=1}^{N}W(\boldsymbol{x}_k;\boldsymbol{\sigma}^{(t)})}$ is the weight for the k-th sample $\boldsymbol{x}_k$, and $\mathcal{L}_d(\boldsymbol{v}_d)$ is the weighted likelihood function for the categorical distribution $h_d(x_d;\boldsymbol{v}_d)$ with data set $\{x_{k,d}\}_{k=1}^N$ and weights $\{w_k\}_{k=1}^N$. Note that when $W(\boldsymbol{x}_k;\boldsymbol{\sigma}^{(t)}), k=1,...,N$ are all equal, $\mathcal{L}_d(\boldsymbol{v}_d)$ degenerates into a standard likelihood function with unit weights, i.e., $w_k = 1, k = 1,...,N$. The analytical solution to the optimization problem in Eq. (5.30) is obtained as

$$\widehat{v}_{d,i}^{(t)} = \frac{1}{N}\sum_{k=1}^{N} w_k \mathbb{I}\{x_{k,d} = s_{d,i}\}, \quad i = 1,...,n_d, \tag{5.31}$$

which coincides with the expression of $\widehat{\boldsymbol{v}}^{(t)}$ in Eq. (5.26). Note that $0 \le \widehat{v}_{d,i} \le 1$, and $\sum_{i=1}^{n_d}\widehat{v}_{d,i}^{(t)} = 1$.

The basic idea of the proposed BiCE method is to employ a Bayesian approach for estimating the parameter vector $\boldsymbol{v}_d$ that aims at avoiding the overfitting problem through adding a regularization term in Eq. (5.28). In particular, the weighted likelihood function in the second line of Eq. (5.28) is multiplied by a prior distribution over the parameter vector $\boldsymbol{v}_d$. Instead of solving the regularized optimization problem, we derive the full posterior predictive distribution for the categorical distribution. The imposed prior distribution acts as an additional information set, and through Bayes' rule, we combine the two sources of information. If one source contains more information than the other, the posterior distribution will be pulled towards it; the relative 'strength' between the prior and the data is adjusted by the α and also the prior parameters.

The prior $f'(\boldsymbol{v_d})$ is chosen to be a Dirichlet distribution in this paper, which is the conjugate prior for the parameter vector $\boldsymbol{v}_d$ of a categorical distribution. The PMF of the Dirichlet distribution can be expressed as follows

$$f'(\boldsymbol{v_d}) = \mathrm{Dir}(\boldsymbol{v}_d;\boldsymbol{\theta}_d) = \frac{1}{B(\boldsymbol{v}_d)}\prod_{i=1}^{n_d}(v_{d,i})^{\theta_{d,i}-1}\mathbb{I}\{\boldsymbol{v}_d \in \mathcal{V}_d\}, \tag{5.32}$$

where $\boldsymbol{\theta}_d = \{\theta_{d,i} > 0\}_{i=1}^{n_d}$ represents the parameter vector of the prior, and $B(\boldsymbol{v}_d)$ is the normalizing constant with $B(\cdot)$ being the multivariate Beta function.

Combining Eq. (5.32) with the weighted likelihood in Eq. (5.30), the posterior distribution $f''(\boldsymbol{v_d})$ is also a Dirichlet distribution. In fact, according to Bayes' rule, $f''(\boldsymbol{v_d})$ can be expressed as

$$\begin{aligned} f''(\boldsymbol{v}_d) &\propto f'(\boldsymbol{v}_d)\mathcal{L}_d(\boldsymbol{v}_d) \\ &\propto \prod_{i=1}^{n_d}(v_{d,i})^{N\widehat{v}_{d,i}^{(t)}+\theta_{d,i}-1}\mathbb{I}\{\boldsymbol{v}_d \in \mathcal{V}_d\} \\ &= \mathrm{Dir}(\boldsymbol{v}_d; N\widehat{\boldsymbol{v}}_d^{(t)} + \boldsymbol{\theta_d}). \end{aligned} \tag{5.33}$$

We then utilize the full distribution of the posterior $f''(\boldsymbol{v}_d)$ for updating the categorical parametric family $h_d(x_d;\boldsymbol{v}_d)$. That is, we calculate the probability $\widetilde{\mu}_{d,i}^{(t)}$ that each state $s_{d,i}$ of $h_d(x_d;\boldsymbol{v}_d)$ occurs under the Dirichlet posterior distribution in Eq. (5.33). Based on the total probability theorem, $\widetilde{\mu}_{d,i}^{(t)}$ can be calculated through

$$\widetilde{\mu}_{d,i}^{(t)} = \int_{\mathcal{V}_d} h_d(s_{d,i};\boldsymbol{v}_d)f''(\boldsymbol{v}_d)d\boldsymbol{v}_d = \int_{\mathcal{V}_d} v_{d,i}\mathrm{Dir}(\boldsymbol{v}_d; N\widehat{\boldsymbol{v}}_d^{(t)} + \boldsymbol{\theta_d})d\boldsymbol{v}_d = \mathbb{E}_{\mathrm{Dir}}[V_{d,i}].$$

Since the mean value of a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{v}_d;\boldsymbol{\theta}_d)$ is explicitly given by $\frac{\theta_{d,i}}{\sum_{j=1}^{n_d}\theta_{d,j}}, i = 1,...,n_d$, [38], we further have

$$\begin{aligned}\widetilde{\mu}_{d,i}^{(t)} &= \frac{N\widehat{v}_{d,i}^{(t)} + \theta_{d,i}}{\sum_{j=1}^{n_d}\left(N\widehat{v}_{d,j}^{(t)} + \theta_{d,j}\right)} \\ &= \frac{N\widehat{v}_{d,i}^{(t)} + \theta_{d,i}}{N + \sum_{j=1}^{n_d}\theta_{d,j}} \\ &= \lambda_d \widehat{v}_{d,i}^{(t)} + (1-\lambda_d)\frac{\theta_{d,i}}{\sum_{j=1}^{n_d}\theta_{d,j}},\end{aligned} \tag{5.34}$$

where $\lambda_d \triangleq \frac{N}{N+\sum_{j=1}^{n_d}\theta_{d,j}}$ denotes the combination factor.

According to Eq. (5.34), this estimator $\widetilde{\mu}_{d,i}^{(t)}$ can be written as a linear combination of the weighted MLE $\widehat{v}_{d,i}^{(t)}$, which exploits the information of the weighted samples, and the prior estimator $\frac{\theta_{d,i}}{\sum_{i=1}^{n_d}(\theta_{d,i})}$, which can explore a different range of the sample space. The relative 'strength' of $\widehat{v}_{d,i}^{(t)}$ is indicated by the combination factor λ_d and is tuned by $\sum_{i=1}^{n_d}\theta_{d,i}$. Evidently, the smaller the $\sum_{i=1}^{n_d}\theta_{d,i}$, the more dominant the $\widehat{v}_{d,i}^{(t)}$, and vice versa. We, therefore, favor a balanced prior with an appropriately large $\sum_{i=1}^{n_d}\theta_{d,i}$, that will not dominate the estimator but can act as a regularizer to deviate the potentially overfitted weighted MLE $\widehat{v}_{d,i}^{(t)}$. A further investigation of the prior distribution is left for future work, and in this paper, we simply employ a symmetric Dirichlet prior for each $\boldsymbol{v}_d$, and set

$$\theta_{d,i} = b; \qquad i = 1,...,n_d, d = 1,...,n, \tag{5.35}$$

where b is the hyperparameter. A natural choice of b is one, which implies an uninformative uniform prior. However, selecting a larger b leads to better results in our numerical investigations. We hence suggest choosing b heuristically as a proportion κ of the sample size per level N so that the resulting combination factor $\lambda_d = \frac{1}{1+\kappa n_d}$ is a positive constant independent of N. The larger κ, the smaller the combination factor λ_d, and hence, the larger the influence of the prior term on $\widetilde{\mu}_{d,i}^{(t)}$. To construct a balanced prior, an appropriate κ is investigated through a parameter study performed in Section 5, where we find that $b = \kappa N = 0.01N$ is a good choice for all investigated cases.

We note that, $\widetilde{\mu}_{d,i}^{(t)}$ converges to $\widehat{v}_{d,i}^{(t)}$ as the sample size N approaches infinity. Considering that $\widehat{v}_{d,i}^{(t)}$ is the normalized IS estimator of the optimal CE parameter $v_{d,i}^{(t,*)}$ in Eq. (5.25), both $\widetilde{\mu}_{d,i}^{(t)}$ and $\widehat{v}_{d,i}^{(t)}$ will converge to $v_{d,i}^{(t,*)}$. Therefore, similar to $\widehat{v}_{d,i}^{(t)}$, the accuracy of $\widetilde{\mu}_{d,i}^{(t)}$ is guaranteed for a large sample size. Conversely, $\widetilde{\mu}_{d,i}^{(t)}$ is positive even for a small sample size. It holds that $\widetilde{\mu}_{d,i}^{(t)} \geq \frac{b}{N+b\cdot n_d} = \frac{\kappa}{1+\kappa\cdot n_d}$. Hence, the 'zero count problem' is less likely to occur.

In this way, $\widetilde{\boldsymbol{\mu}}_d^{(t)} = (\widetilde{\mu}_{d,1}^{(t)},...,\widetilde{\mu}_{d,n_d}^{(t)})$ forms a new parameter vector for the one-dimensional categorical family $h_d(x_d;\boldsymbol{v}_d)$. $h_d(x_d;\widetilde{\boldsymbol{\mu}}_d^{(t)})$ is also known as the posterior predictive distribution in Bayesian statistics, so we term $\widetilde{\boldsymbol{\mu}}_d^{(t)}$ the Bayesian estimator of $\boldsymbol{v}_d$. After obtaining the Bayesian estimator $\widetilde{\boldsymbol{\mu}}_d^{(t)}$ for each dimension $d = 1,...,n$ through Eq. (5.34), we concatenate the results to get the Bayesian estimator $\widetilde{\boldsymbol{\mu}}^{(t)}$ for the independent categorical distribution $h(\boldsymbol{x};\boldsymbol{v})$, i.e., $\widetilde{\boldsymbol{\mu}}^{(t)} = [\widetilde{\boldsymbol{\mu}}_1^{(t)};...;\widetilde{\boldsymbol{\mu}}_d^{(t)}]$. The posterior predictive distribution $h(\boldsymbol{x};\widetilde{\boldsymbol{\mu}}^{(t)})$ is then employed as the reference distribution $p_{ref}(\boldsymbol{x})$

for the $(t+1)$-th iteration in iCE. The resulting algorithm is termed the Bayesian improved cross entropy method (BiCE) and is given in Algorithm 7.

It should be stressed that the BiCE estimator $\widehat{p}_f$ is theoretically unbiased since

$$\mathbb{E}[\widehat{p}_f] = \mathbb{E}[\mathbb{E}[\widehat{p}_f|\widetilde{\boldsymbol{\mu}}^{(T)}]] = \mathbb{E}[p_f] = p_f. \tag{5.36}$$

$\widetilde{\boldsymbol{\mu}}^{(T)}$ collects the parameters of the posterior predictive distribution $h(\boldsymbol{x};\widetilde{\boldsymbol{\mu}}^{(T)})$ in the final T-th level, or the final IS distribution. Different from the standard iCE method, where these parameters can converge to zero due to the zero count problem (or overfitting), they are guaranteed to be positive in BiCE. Hence, each state in the failure domain can be reached by the IS distribution obtained by BiCE, and this IS estimator is unbiased, i.e., it holds $\mathbb{E}[\widehat{p}_f|\widetilde{\boldsymbol{\mu}}^{(T)}] = p_f$.

Algorithm 7: Bayesian improved cross entropy algorithm

Input: N, δ_{tar}, δ_ϵ, b, auxiliary LSF $g_a(\boldsymbol{x})$, input distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$

$t \leftarrow 1$, $t_{max} \leftarrow 50$, $\sigma_0 \leftarrow \infty$

$h(\boldsymbol{x};\widetilde{\boldsymbol{\mu}}^{(t-1)}) \leftarrow p_{\boldsymbol{X}}(\boldsymbol{x})$

while *true* **do**

  Generate N samples $\{\boldsymbol{x}_k\}_{k=1}^N$ from $h(\boldsymbol{x};\widetilde{\boldsymbol{\mu}}^{(t-1)})$ and calculate the corresponding LSF values $\{g_a(\boldsymbol{x}_k)\}_{k=1}^N$

  Compute the sample c.o.v. $\widehat{\delta}$ of $\left\{\frac{\mathbb{I}\{g_a(\boldsymbol{x}_k)\leq 0\}}{\Phi(-g_a(\boldsymbol{x}_k)/\sigma^{(t-1)})}\right\}_{k=1}^N$

  if $t > t_{max}$ *or* $\widehat{\delta} \leq \delta_\epsilon$ **then**

    Break

  Determine $\sigma^{(t)}$ through solving Eq. (5.17) using the alternative weight function defined in Eq. (5.23)

  Calculate $W(\boldsymbol{x}_i)$ for each $i = 1, ..., N$ through Eq. (5.16)

  Compute $\widetilde{\mu}_{d,i}^{(t)}$ through Eq. (5.34), for each d and i with $\theta_{d,i}$ given by Eq. (5.35)

  $t \leftarrow t+1$

$T \leftarrow t-1$

Use $h(\boldsymbol{x};\widetilde{\boldsymbol{\mu}}^{(T)})$ as the IS distribution and calculate the IS estimator $\widehat{p}_f$ through Eq. (5.2)

Output: $\widehat{p}_f$

Remark 5.4.1. *Instead of using the full posterior distribution, one can also utilize the mode of the posterior distribution $\widetilde{\boldsymbol{v}}_d^{(t)}$ as a point estimate of $\boldsymbol{v}_d$, which is known as the maximum a posteriori (MAP) estimator in Bayesian statistics. By definition, the MAP estimator can be expressed as*

$$\widetilde{\boldsymbol{v}}_d^{(t)} = \arg\max_{\boldsymbol{v}_d \in \mathcal{V}_d} f''(\boldsymbol{v}_d). \tag{5.37}$$

Substituting the posterior $f''(\boldsymbol{v}_d)$ with the expression in Eq. (5.33) and then solving the optimization problem in Eq. (5.37) with the Lagrange multiplier method gives us

$$\begin{aligned}\widetilde{v}_{d,i}^{(t)} &= \frac{N\widehat{v}_{d,i}^{(t)} + \theta_{d,i} - 1}{N\sum_{i=1}^{n_d}\widehat{v}_{d,i}^{(t)} + \sum_{i=1}^{n_d}(\theta_{d,i}-1)} \\ &= \frac{N}{N+\sum_{i=1}^{n_d}(\theta_{d,i}-1)}\widehat{v}_{d,i}^{(t)} + \frac{\sum_{i=1}^{n_d}(\theta_{d,i}-1)}{N+\sum_{i=1}^{n_d}(\theta_{d,i}-1)}\frac{\theta_{d,i}-1}{\sum_{i=1}^{n_d}(\theta_{d,i}-1)}.\end{aligned} \tag{5.38}$$

Comparing Eq. (5.38) and Eq. (5.34), we find that the MAP estimator $\widetilde{v}_{d,i}^{(t)}$ with prior parameter $\theta_{d,i} > 1$ is the same as the Bayesian estimator $\widetilde{\mu}_{d,i}^{(t)}$ with prior parameter $\theta_{d,i} - 1$. When $\theta_{d,i} = 1$, the MAP estimator $\widetilde{v}_{d,i}^{(t)}$ reduces to the weighted MLE $\widehat{v}_{d,i}^{(t)}$.

Remark 5.4.2. *If the symmetric Dirichlet prior described in Eq. (5.35) is applied in the BiCE method, it holds that $\widetilde{\mu}_{d,i}^{(t)} = \frac{N}{N+b\cdot n_d}\widehat{v}_{d,i}^{(t)} + \frac{b}{N+b\cdot n_d}$ according to Eq. (5.34), which means the probability that X_d equals $s_{d,i}$ in the posterior predictive distribution is at least $\frac{b}{N+b\cdot n_d}$ (or $\frac{\kappa}{1+\kappa\cdot n_d}$ when b is chosen as a minor proportion κ of N). To avoid this probability being close to zero, the number of states for X_d, n_d, should not be too large.*

Remark 5.4.3. *Although derived in the context of the iCE method, the Bayesian estimator $\widetilde{\mu}_{d,i}^{(t)}$ can also be combined with other CE-based methods for obtaining an unbiased network reliability estimator.*

5.5 Examples

For testing the performance of the BiCE method, we investigate three numerical examples: a toy example with algebraic LSF, a multi-state two-terminal reliability problem, and a direct current (DC) power flow problem for the IEEE39 benchmark. We compare the proposed BiCE method with the iCE method in terms of their relative efficiency with respect to crude MCS, a concept that is borrowed from statistics [32]. The relative efficiency measure reads as follows

$$\mathrm{relEff}(\widehat{p}_f) \triangleq \frac{p_f \cdot (1 - p_f)}{\mathrm{MSE}(\widehat{p}_f) \times \mathrm{Cost}(\widehat{p}_f)}, \tag{5.39}$$

where $\mathrm{MSE}(\widehat{p}_f)$ represents the mean square error of the failure probability estimator $\widehat{p}_f$, and the cost of an algorithm is measured by the number of evaluations of the LSF. Note that the relative efficiency of crude MCS is always equal to one.

5.5.1 Parameter study: system with linear limit state functions

In this example, we consider an LSF $g_1(\boldsymbol{x})$, which is a linear combination of 50 random variables. The coefficients of the first and the last 10 random variables are set to be 2 and 0, respectively, while for the remaining random variables, the coefficients are fixed at 1. The LSF reads

$$g_1(\boldsymbol{x}) = \sum_{d=1}^{10} 2 \cdot x_d + \sum_{d=11}^{40} 1 \cdot x_d + \sum_{d=41}^{50} 0 \cdot x_d. \tag{5.40}$$

5.5.1.1 Binary input

We first assume that $\{X_d\}_{d=1}^{50}$ are independent and identically distributed (i.i.d.) Bernoulli random variables with success probability 10^{-3}, i.e., the probability that each X_d takes the value 1 is 10^{-3}.

We estimate the probability that $g_1(\boldsymbol{X}) \geq 6$ using the BiCE and compare the result with that of the standard iCE approach. The exact solution is $1.387 \cdot 10^{-7}$ through the convolution of two binomial distributions. We fix the sample size N at 500 and 2,000 for BiCE and iCE, respectively, and set $\delta_{tar} = \delta_\epsilon = 1$ for both methods. 5,000 repeated runs of each estimator are carried out to calculate the relative bias, the sample c.o.v., and the average computational cost (i.e., the average number of calls of the LSF) of the estimator. Also, the influence of different prior parameters on the performance of BiCE is investigated. We apply the symmetric Dirichlet prior defined in Eq. (5.35) and vary the parameter b therein. We note that when $n_d = 2$ the Dirichlet distribution degenerates into the Beta distribution. The results are summarized in Table 5.1.

Table 5.1: Performance of BiCE for Example 5.1.1.

method	BiCE	BiCE	BiCE	BiCE	BiCE	iCE
sample size, N	500	500	500	500	500	2,000
prior parm., b	1	5	10	25	50	/
relative bias	0.012	0.011	0.007	0.007	0.006	-0.375
sample c.o.v.	0.196	0.122	0.155	0.287	0.787	0.372
comp. cost	4,660	4,127	3,555	2,495	1,500	19,967
relEff ($\times 10^4$)	4.01	11.64	8.42	3.51	0.78	0.13

We can see from the table that the iCE method, even with $N = 2,000$ samples per level, which is four times larger than the number of samples used with the BiCE, performs poorly with a strong negative bias. This is due to the zero count problem described in Subsection 5.4.2; the failure states of some of the input random variables are ignored throughout the sampling process, which leads to an under-representation of the failure domain. By contrast, BiCE with an uninformative prior $b = 1$, works well, resulting in an efficient yet accurate estimator. The performance of the BiCE can be further enhanced through employing a larger b. The prior with $b = 5 = 0.01N$ is the optimal choice in this example for which the efficiency of BiCE is $1.16 \cdot 10^5$ times larger than that of crude MCS. Selecting a larger value of b leads to poorer results.

To further illustrate the zero count problem, Fig. 5.2 shows the influence of the number of samples per level, N, on the c.o.v. and the relative bias of the BiCE and iCE estimates. In this figure, the blue solid line represents the BiCE method with uninformative prior and target c.o.v. $\delta_{tar} = \delta_\epsilon = 1$; the red dashed line shows the BiCE method with uninformative prior and target c.o.v. $\delta_{tar} = \delta_\epsilon = 1.5$. Both variants show a negligible relative bias for all considered N. In contrast, the relative bias of the standard iCE method is almost -100% when the number of samples is small, i.e., $N = 500$, and gradually approaches 0 as N increases. Obviously, the zero count problem is less likely to happen for a larger number of samples per intermediate level.

5.5.1.2 Multi-state input

Next, we assume that $\{X_d\}_{d=1}^{50}$ follows the i.i.d categorical distribution with categories 0, 1, and 3. The probabilities assigned to these categories are 0.899, 0.1, and 10^{-3}. In this subsection, we estimate

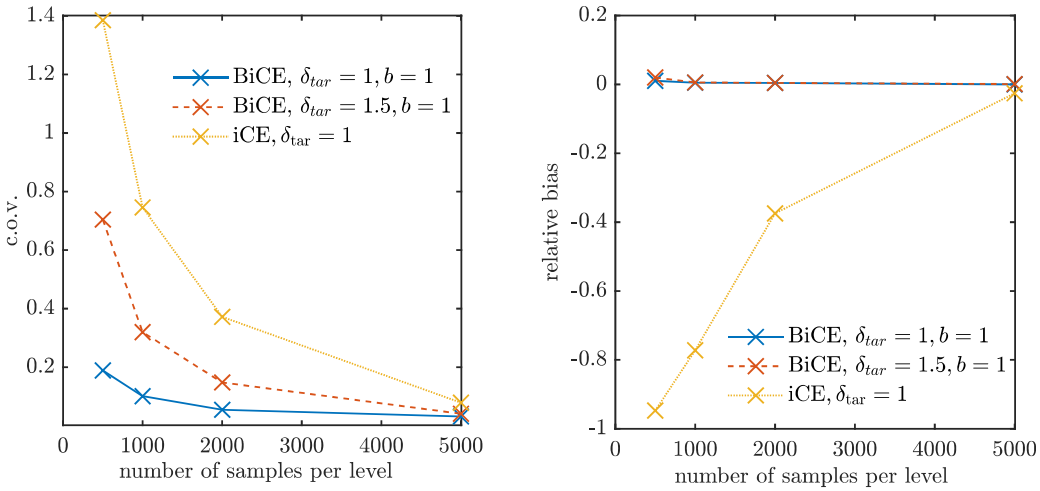


Figure 5.2: Parameter study for Example 5.1.1. (left) c.o.v. (right) relative bias.

the probability that $g_1(\boldsymbol{x}) \geq 19$. The exact value of this probability is approximated through crude MCS with 10^7 samples, resulting in $7.3 \cdot 10^{-5}$. 5000 repeated runs of BiCE with hyperparameters $\delta_{tar} = \delta_\epsilon = 1$ and $N = 1,000$ are performed. The obtained results are then compared with the standard iCE approach with $\delta_{tar} = \delta_\epsilon = 1$ and $N = 2,000$, and are summarized in Table 5.2. Similarly to the binary case in Subsection 5.1.1, the BiCE with the uniform prior, that is, the case where $b = 1$, outperforms the standard iCE approach, and the performance can be further improved through applying a larger b. In this example, $b = 0.01N = 10$ appears to be a good choice.

Fig. 5.3 shows the performance of the iCE and the BiCE methods for varying the number of samples per intermediate level for multi-state input. Similarly to Fig. 5.2, which is given for binary input, the relative bias of the iCE method approaches zero as N goes from 500 to 5,000.

Table 5.2: Performance of BiCE for Example 5.1.2.

method	BiCE	BiCE	BiCE	BiCE	BiCE	iCE
sample size, N	1,000	1,000	1,000	1,000	1,000	2,000
prior parm., b	1	10	20	50	100	/
relative bias	-0.004	-0.014	-0.054	-0.047	-0.047	-0.375
sample c.o.v.	0.285	0.107	0.132	0.254	0.684	0.124
comp. cost	7,484	5,997	4,999	3,000	2,000	15,966
relEff	22.529	157.171	134.71	68.428	14.570	5.642

5.5.1.3 Adaptive selection of $\sigma^{(t)}$ of intermediate target distributions

Fig. 5.4 shows the adaptive selection of $\sigma^{(t)}$ of intermediate target distributions in the BiCE method. We adopt $b = 1, \delta_{tar} = \delta_\epsilon = 1$ and discuss four different settings: binary input in Eq. (5.40) with

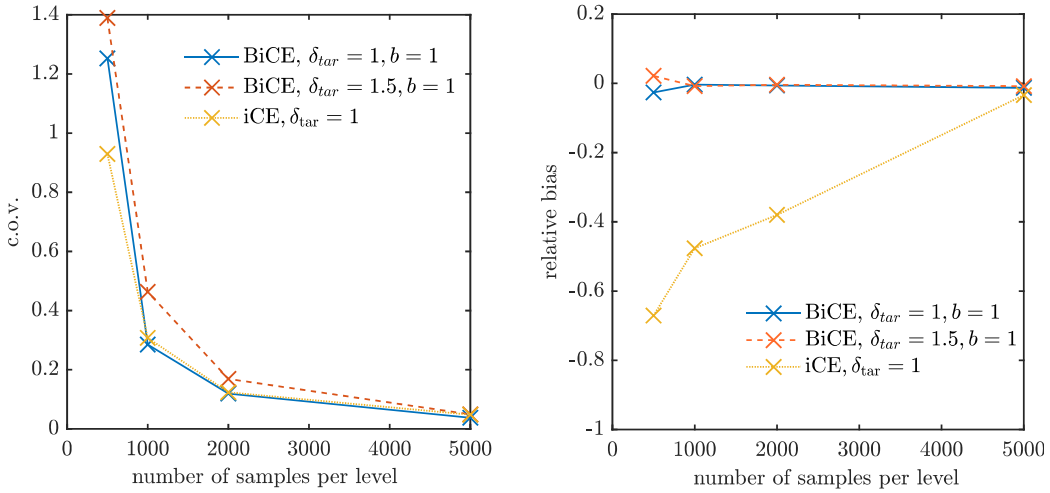


Figure 5.3: Parameter study for Example 5.1.2. (left) c.o.v. (right) relative bias.

moderate N, multi-state input with moderate N, binary input with large N and multi-state input with large N. The ordinate in each subfigure represents the sample c.o.v. of the weight $W^{alt}(\boldsymbol{X};\sigma)$ defined in Eq. (5.23), which is a function of σ at each t-th intermediate level. These functions are depicted in colored solid lines in the subfigure.

In the first few intermediate levels, the weight $W^{alt}(\boldsymbol{X};\sigma)$ is highly skewed for a small σ, and hence, the sample c.o.v. derived from a limited number of samples can be significantly biased. We observe a notable increase of the sample c.o.v. when increasing the sample size.

For higher levels, however, the sample c.o.v. is more robust to the change in the sample size. The sample c.o.v. is a strictly decreasing function of σ and has a unique intersection with the dotted line, the target c.o.v., δ_{tar}, which further indicates a unique solution $\sigma^{(t)}$ to Eq. (5.17) at each t-th level. This behavior is consistent with Theorems 4.1 and 4.2. Each $\sigma^{(t)}$ corresponds to a grey vertical line in the subfigure. We can see that the adaptation of $\sigma^{(t)}$ is similar for both moderate and large sample settings in this example.

5.5.2 Multi-state two-terminal reliability

Fig.5.5 shows the topology of a multi-state two-terminal network with 11 nodes and 20 edges (or arcs), which is motivated by the third example of [42]. The capacity of each edge, that is, the maximum flow that can pass through the edge, takes the same three states $0, 3$, and 5 with probability 10^{-3}, 0.1, and 0.899, respectively. Also, the states of each edge are independent. We are required to estimate the probability that the maximum flow from the source node s to the sink node t is less or equal to a predefined demand $D_{tar} = 6$. The true value of the probability is approximately $1.8 \cdot 10^{-4}$ according to the crude MCS results (with sample size $2 \cdot 10^6$), and this value is employed to assess the accuracy of the proposed BiCE approach. We choose $\delta_{tar} = \delta_\epsilon = 1$ and $N = 1,000$ and use an uninformative uniform prior, $\text{Dir}(\cdot;\boldsymbol{\theta} = [1,1,1]^T)$ for each dimension in BiCE, i.e., we set $b = 1$. The relative bias, sample c.o.v., average computational cost, and relative efficiency of 200 repeated runs of the BiCE are -0.004, 0.100, 9,385, and 59.091, respectively, indicating an efficient yet accurate

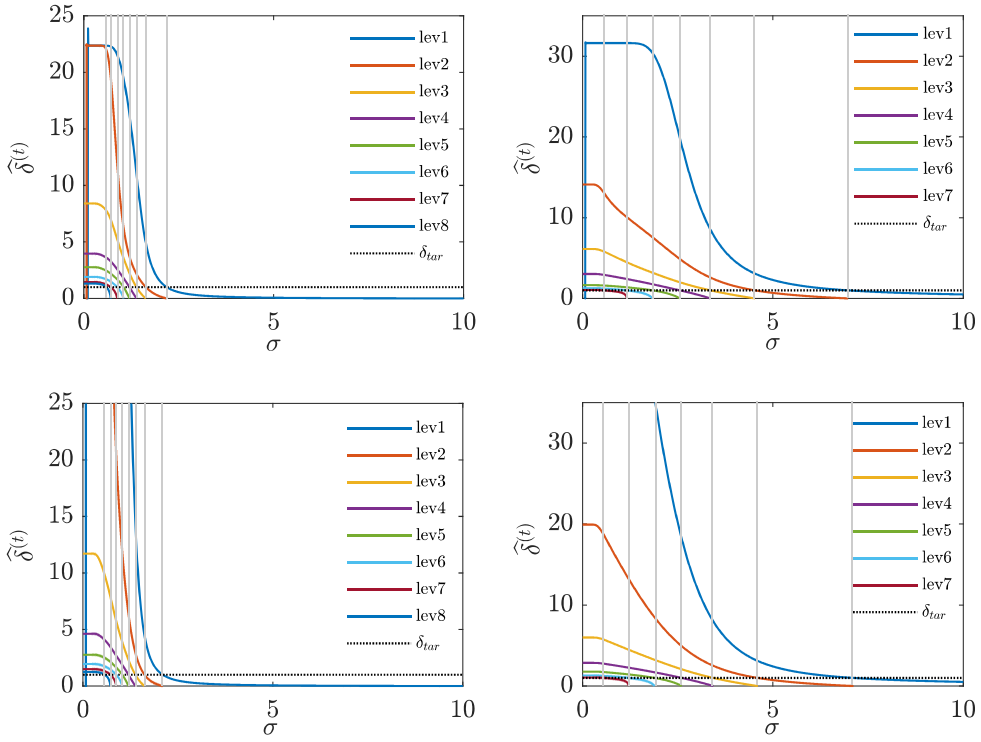


Figure 5.4: Adaptive selection of $\sigma^{(t)}$ in the BiCE method with $b = 1, \delta_{tar} = \delta_\epsilon = 1$. (left top) binary input, $N = 500$. (right top) multi-state input, $N = 1000$. (left bottom) binary input, $N = 10^5$. (right bottom) multi-state input, $N = 10^5$.

estimator. In contrast, the standard iCE with hyperparameters $\delta_{tar} = \delta_\epsilon = 1$ and $N = 2,000$ will seriously underestimate the failure probability. On the other hand, the performance of the BiCE estimator can be improved through setting $b = 0.01N = 10$. These results are summarized in Table 5.3. Table 5.4 shows the PMF of the final IS distribution in BiCE averaged over 200 repetitions. We can see from this table that the IS distribution of the 14-th edge differs the most from the input distribution in BiCE, followed by edges 16/17 and edges 4/13/19. This indicates that these are the most important edges for the failure probability. For the remaining edges, the IS distribution differs only slightly from the input distribution.

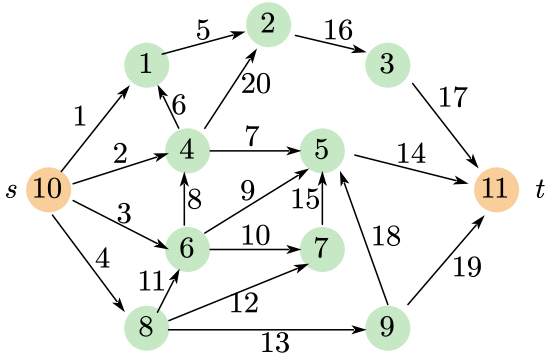


Figure 5.5: Topology of the network for example 5.2.

Table 5.3: Performance of BiCE for Example 5.2.

method	BiCE	BiCE	iCE
sample size, N	$1,000$	$1,000$	$2,000$
prior parm., b	1	10	/
relative bias	-0.004	-0.021	-0.253
sample c.o.v.	0.100	0.089	0.168
comp. cost	$9,385$	$8,650$	$20,400$
relEff	59.091	76.793	2.952

Table 5.4: The PMF of the IS distribution in BiCE for Example 5.2 (averaged over 200 repetitions).

state	edge 14	edge 16	edge 17	edge 4	edge 13	edge 19	the rest
0	0.334	0.170	0.179	0.128	0.123	0.127	≈ 0.002
3	0.628	0.355	0.348	0.254	0.254	0.250	≈ 0.10
5	0.037	0.475	0.473	0.618	0.623	0.624	≈ 0.898

5.5.3 Power transmission network with cascading failure

In this example, we consider the IEEE39 benchmark system, a simplified model of the high voltage transmission system in the northeast of the U.S.A. The model was first presented in 1970 [7] and has been extensively used as a benchmark model in power system analysis [2, 47, 43].

It consists of 39 buses, including 10 generators and 19 load buses, 34 transmission lines, and 12 transformers. The topology of the network is illustrated in Fig. 5.6 where all the buses are modeled as nodes and transmission lines together with transformers are modeled as edges, so there are in total 39 nodes and 46 edges in the model. In the figure, orange circles stand for the source nodes, representing the 10 generators, and grey circles represent the terminal nodes, the 19 load buses. Edges are weighted by their reactance values shown on the right-hand side of Fig. 5.6 and by their capacities shown on the left-hand side.

Through solving the DC load flow problem described in the literature (e.g. [24] for IEEE39 benchmark model), one can derive the actual DC flow that passes through each edge of the network. An edge fails when the DC flow exceeds its capacity, and the initial edge failures change the topology of the network, resulting in a new configuration of the flow across the remaining components, which in turn may lead to further overloading of the edge. This phenomenon is also known as cascading failure and is modeled here based on [17]. The system will finally reach an equilibrium state where no further edges are overloaded. In general, only a part of the original power demand at the load buses (the terminal nodes) can be matched in the equilibrium.

We assume that nodes will never fail, and the state of each edge follows an i.i.d Bernoulli distribution,

with component failure probability 10^{-3}. The LSF is then defined as a function of the system state $\boldsymbol{x}$, which is a binary vector, as follows:

$$g_3(\boldsymbol{x}) = 30\% - L(\boldsymbol{x}), \tag{5.41}$$

where $L(\boldsymbol{x})$ denotes the percentage of the original power demand that cannot be matched when the system achieves the equilibrium after the cascading failure, and the failure probability is defined as the probability of such percentage loss being greater or equal than the threshold, 30%.

A crude MCS procedure with 10^6 samples is used to validate the results of the proposed method, which gives a failure probability of $9.3 \cdot 10^{-5}$. We then apply the BiCE algorithm and set the hyperparameters $N = 1,000, \delta_{tar} = \delta_\epsilon = 1.5$. Two different kinds of prior distributions are considered here, the uninformative prior $\text{Beta}(\cdot; \boldsymbol{\theta} = [1,1]^T)$ and an informative prior $\text{Beta}(\cdot; \boldsymbol{\theta} = [10,10]^T)$, which correspond to setting $b = 1$ and $b = 10$, respectively. The second and third column of Table 5.5 shows the performance of the BiCE methods after 500 repeated runs. The efficiency of BiCE with the uninformative prior ($b = 1$) is 5.5 times higher than that of crude MCS and is also higher than that of iCE. With an informative Dirichlet prior ($b = 10$), the efficiency is 105 times higher than that of MCS. By contrast, the iCE estimate is significantly biased and has much lower efficiency.

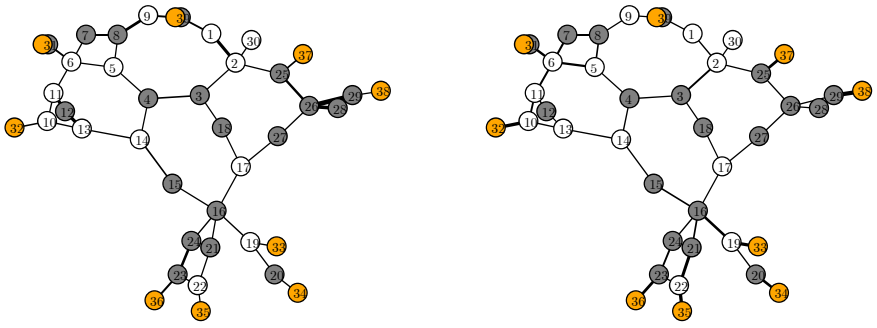


Figure 5.6: IEEE39 bus system, with edge thicknesses proportional to their capacities (left) and reactances (right).

Table 5.5: Performance of BiCE for Example 5.3.

method	BiCE	BiCE	iCE
sample size, N	1,000	1,000	2,000
prior parm., b	1	10	/
relative bias	0.0049	-0.009	-0.409
sample c.o.v.	0.5836	0.143	0.305
comp. cost	5,746	5,000	11,870
relEff	5.494	104.741	3.480

5.6 Conclusions

This paper studies the cross entropy (CE) methods in the context of network reliability assessment. We distinguish three distributions involved in the CE procedure: the optimal importance sampling (IS) distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$, the suboptimal IS distribution $h(\boldsymbol{x};\boldsymbol{v}^*)$, and the chosen IS distribution $h(\boldsymbol{x};\widehat{\boldsymbol{v}})$. Given a certain parametric family, the 'distance' between $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ and $h(\boldsymbol{x};\boldsymbol{v}^*)$ is fixed, and the objective of the CE method is to find a good estimator $\widehat{\boldsymbol{v}}$ that is close to the optimal but inaccessible CE parameter $\boldsymbol{v}^*$. For parametric models that belong to the exponential family, $\widehat{\boldsymbol{v}}$ is the self-normalized IS estimator of the optimal CE parameter $\boldsymbol{v}^*$, and hence converges to $\boldsymbol{v}^*$ as the sample size goes to infinity. Moreover, we show that $\hat{\boldsymbol{v}}$ can be viewed as the solution of a weighted maximum likelihood estimator (MLE) problem given the samples obtained at a certain level of the adaptive CE sampling process. In network reliability assessments with discrete multi-state inputs, the parametric family can be chosen as the independent categorical distribution. In these approaches, the CE estimator $\widehat{\boldsymbol{v}}$ suffers from the 'zero count problem,' which is essentially the overfitting issue, resulting in a poor IS estimator with a strong negative bias. This paper derives the posterior predictive distribution $h(\boldsymbol{x};\widetilde{\boldsymbol{\mu}})$ to update the categorical model instead of the original maximum likelihood estimation $\hat{\boldsymbol{v}}$. By introducing the symmetric Dirichlet prior as shown in Eq. (5.35), the probability assigned to each d-th category of the parametric model is at least $\frac{b}{b\cdot n_d+N}$, where b is the hyperparameter. Hence, the 'zero count problem' is less likely to occur. The Bayesian estimator $\widetilde{\boldsymbol{\mu}}$ is consistent, i.e., $\widetilde{\boldsymbol{\mu}}$ converges to $\boldsymbol{v}^*$ as the sample size goes infinity. Combining the Bayesian estimator $\widetilde{\boldsymbol{\mu}}$ with the standard improved CE (iCE) procedure, a modified iCE method called Bayesian improved cross entropy (BiCE) method is proposed for network reliability analysis, which is theoretically unbiased. The efficiency and accuracy of the proposed method are illustrated through a set of numerical examples, from which it is found that BiCE with an appropriately chosen informative prior can significantly enhance the performance of the iCE method. Our numerical investigations indicate that a uniform prior performs only sub-optimally. In all examples, we observed significantly better performance with a symmetric, informative prior with $b = 0.01N$. It should also be stressed that the BiCE estimator can be skewed, and this is probably due to the limited capacity of the parametric model because of its assumption of independence. If the suboptimal IS distribution $h(\boldsymbol{x};\boldsymbol{v}^*)$ itself is far from the optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$, independent on close $\widetilde{\boldsymbol{\mu}}$ is to $\boldsymbol{v}^*$, the resulting IS estimator is bound to perform poorly

5.7 Acknowledgment

The first author gratefully acknowledges the financial support of the China Scholarship Council.

5.A Self-normalized importance sampling and cross entropy method

In this Appendix, we first introduce the self-normalized IS estimator for estimating the expectation of a general function $H(\boldsymbol{x})$ and then prove that, in the CE (or iCE) method, the chosen parameter vector $\hat{\boldsymbol{v}}$ is the self-normalized IS estimator of the sub-optimal vector $\boldsymbol{v}^*$ for the exponential parametric family.

5.A.1 Self-normalized importance sampling

We consider the following expectation of a general function $H(x)$:

$$\mu = \mathbb{E}_\pi[H(\boldsymbol{X})]. \tag{42}$$

The input distribution $\pi(\boldsymbol{x})$ is only known pointwise to an unknown constant Z. That is

$$\pi(\boldsymbol{x}) = \frac{1}{Z}\pi_u(\boldsymbol{x}). \tag{43}$$

$\pi_u(\boldsymbol{x})$ is the unnormalized form of π. In such case, the standard IS estimator of Eq. (5.2) cannot be applied since the likelihood ratio L, which is defined as the ratio of the input distribution $\pi(\boldsymbol{x})$ to the IS distribution $p_{IS}(\boldsymbol{x})$, is intractable. Instead, the following self-normalized IS estimator can be applied

$$\bar{\mu} = \sum_{k=1}^{N} \frac{W(\boldsymbol{x}_k)}{\sum_k W(\boldsymbol{x}_k)} H(\boldsymbol{x}_k), \tag{44}$$

where $W(\boldsymbol{x}_k) \triangleq \frac{\pi_u(\boldsymbol{x}_k)}{p_{IS}(\boldsymbol{x}_k)}$. It can be proved that the self-normalized IS estimator is consistent, i.e., the estimator converges to the exact value as N goes infinity, under the condition that the sample space of the input distribution $\pi(\boldsymbol{x})$ is included in that of the IS distribution $p_{IS}(\boldsymbol{x})$ [39]. The estimator of Eq. (44) can be less efficient than the standard Monte Carlo estimator that samples directly from $\pi(\boldsymbol{x})$. The efficiency of the self-normalized estimator with respect to the crude Monte Carlo estimator can be measured by the so-called effective sample size (ESS) [29]. ESS represents the number of samples that a crude MCS would need in order to yield the same variance as that of the self-normalized IS estimator of Eq. (44). The ESS can be approximated through the following expression [29]

$$ESS \approx \frac{N}{1 + \delta^2(W(\boldsymbol{X}))}, \quad \boldsymbol{X} \sim p_{IS}(\boldsymbol{x}), \tag{45}$$

where $\delta(W(\boldsymbol{X}))$ represents the coefficient of variation of the weights $W(\boldsymbol{X})$ in Eq. (44), and N is the sample size of the self-normalized IS estimator of Eq. (44).

5.A.2 Cross entropy method with exponential parametric family

In this subsection, we aim at finding a distribution from the exponential family $h(\boldsymbol{x};\boldsymbol{v})$ that has the minimal KL divergence with respect to the distribution π of Eq. (43). Note that the optimal IS distribution $p^*_{\boldsymbol{X}}(\boldsymbol{x})$ in Eq. (5.3) and the intermediate target distribution $p^{(t)}$ in Eq. (5.13) (or Eq. (5.10)) can be regarded as special cases of π, with π_u set equal to $p_{\boldsymbol{X}}(\boldsymbol{x})\mathbb{I}\{g(\boldsymbol{x}) \leq 0\}$ and $p_{\boldsymbol{X}}(\boldsymbol{x})\Phi(-g(\boldsymbol{x})/\sigma^{(t)})$ (or $p_{\boldsymbol{X}}(\boldsymbol{x})\mathbb{I}\{g(\boldsymbol{x}) \leq \gamma^{(t)}\}$), respectively.

The corresponding CE optimization problem is given as

$$\boldsymbol{v}^* = \arg\max_{\boldsymbol{v}\in\mathcal{V}} \sum_{\boldsymbol{x}\in\Omega_X} \pi_u(\boldsymbol{x}) \ln(h(\boldsymbol{x};\boldsymbol{v})) \tag{46}$$

where $\Omega_{\boldsymbol{X}}$ is the sample space of $\boldsymbol{X}$. The summation in Eq. (46) is substituted with the integral for continuous $\boldsymbol{X}$. The sample counterpart of Eq. (46) reads

$$\widehat{\boldsymbol{v}} = \arg\max_{\boldsymbol{v}\in\mathcal{V}} \frac{1}{N} \sum_{k=1}^{N} \frac{\pi_u(\boldsymbol{x}_k)}{p_{ref}(\boldsymbol{x}_k)} \ln(h(\boldsymbol{x}_k;\boldsymbol{v})), \qquad \boldsymbol{x}_k \sim p_{ref}(\cdot). \tag{47}$$

The exponential family of distributions is defined as the collection of distributions that have the following general form:

$$f(\boldsymbol{x};\boldsymbol{\eta}) = a(\boldsymbol{x})\exp(\boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{t}(\boldsymbol{x}) - A(\boldsymbol{\eta})), \tag{48}$$

where $\boldsymbol{\eta} = (\eta_1, ..., \eta_m)^{\mathrm{T}}$ is often referred to as the canonical parameter. The statistic $\boldsymbol{t}(\boldsymbol{x}) = (t_1(\boldsymbol{x}), ..., t_n(\boldsymbol{x}))^{\mathrm{T}}$ is referred to as the sufficient statistic. The function $A(\boldsymbol{\eta})$ is known as the cumulant function.

In the following, we reparameterize the exponential family with $\boldsymbol{v} = \nabla_{\boldsymbol{\eta}} A(\boldsymbol{\eta})$. Through inserting $h(\boldsymbol{x};\boldsymbol{v})$ into Eq. (46) and setting the gradient of the objective function equal to zero, we get $\boldsymbol{v}_c^* = \mathbb{E}_\pi[\boldsymbol{t}(\boldsymbol{X})]$ [30]. Typically, $\boldsymbol{v}_c^*$ satisfies the constraint $\boldsymbol{v} \in \mathcal{V}$, and we have $\boldsymbol{v}^* = \boldsymbol{v}_c^*$. Note that the explicit expression of $\boldsymbol{v}^*$ depends on a prior knowledge of the distribution of $\pi(\boldsymbol{x})$, or equivalently a knowledge of the unknown constant Z and therefore cannot be directly used. In such case, the sample counterpart Eq. (47) is solved instead, which gives us $\widehat{\boldsymbol{v}} = \frac{\sum_{k=1}^N W(\boldsymbol{x_k})\boldsymbol{t}(\boldsymbol{x}_k)}{\sum_{k=1}^N W(\boldsymbol{x_k})}$, where $W(\boldsymbol{x}_k) = \frac{\pi_u(\boldsymbol{x}_k)}{p_{ref}(\boldsymbol{x}_k)}$.

According to Eq. (44), $\widehat{\boldsymbol{v}}$ is the self-normalized estimator of $\boldsymbol{v}^*$ with $H(\boldsymbol{x})$ being the sufficient statistic $\boldsymbol{t}(\boldsymbol{x})$. The accuracy of the estimator $\widehat{\boldsymbol{v}}$ can be measured by the ESS defined in Eq. (45).

References

[1] C. Alexopoulos and G. S. Fishman. "Characterizing stochastic flow networks using the Monte Carlo method". In: *Networks* 21.7 (1991), pp. 775–798.

[2] T. Athay, R. Podmore, and S. Virmani. "A practical method for the direct analysis of transient stability". In: *IEEE Transactions on Power Apparatus and Systems* 2 (1979), pp. 573–584.

[3] S.-K. Au and J. L. Beck. "A new adaptive importance sampling scheme for reliability calculations". In: *Structural Safety* 21.2 (1999), pp. 135–158.

[4] S.-K. Au and J. Beck. "Important sampling in high dimensions". In: *Structural Safety* 25.2 (2003), pp. 139–163.

[5] M. O. Ball, C. J. Colbourn, and J. S. Provan. "Network reliability". In: *Handbooks in Operations Research and Management Science* 7 (1995), pp. 673–762.

[6] J. Behrensdorf, T.-E. Regenhardt, M. Broggi, and M. Beer. "Numerically efficient computation of the survival signature for the reliability analysis of large networks". In: *Reliability Engineering & System Safety* 216 (2021), p. 107935.

[7] G. Bills. *On-line stability analysis study, RP 90-1*. Tech. rep. North American Rockwell Information Systems Co., Anaheim, CA (USA), 1970.

[8] Z. I. Botev, P. L'Ecuyer, G. Rubino, R. Simard, and B. Tuffin. "Static network reliability estimation via generalized splitting". In: *INFORMS Journal on Computing* 25.1 (2013), pp. 56–71.

[9] Z. I. Botev, P. L'Ecuyer, and B. Tuffin. "Reliability estimation for networks with minimal flow demand and random link capacities". In: *arXiv preprint arXiv:1805.03326* (2018).

[10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[11] S. Bulteau and M. El Khadiri. "A new importance sampling Monte Carlo method for a flow network reliability problem". In: *Naval Research Logistics (NRL)* 49.2 (2002), pp. 204–228.

[12] J.-E. Byun and J. Song. "Generalized matrix-based Bayesian network for multi-state systems". In: *Reliability Engineering & System Safety* 211 (2021), p. 107468.

[13] H. Cancela and M. El Khadiri. "The recursive variance-reduction simulation algorithm for network reliability evaluation". In: *IEEE Transactions on Reliability* 52.2 (2003), pp. 207–212.

[14] J. Chan, I. Papaioannou, and D. Straub. "An adaptive subset simulation algorithm for system reliability analysis with discontinuous limit states". In: *Reliability Engineering & System Safety* 225 (2022), p. 108607.

[15] J. Chan, I. Papaioannou, and D. Straub. "Improved cross-entropy-based importance sampling for network reliability assessment". In: *Proceedings of the 13th International Conference on Structural Safety & Reliability*. ICOSSAR. 2022.

[16] J. C. Chan and D. P. Kroese. "Improved cross-entropy method for estimation". In: *Statistics and Computing* 22.5 (2012), pp. 1031–1040.

[17] P. Crucitti, V. Latora, and M. Marchiori. "Model for cascading failures in complex networks". In: *Physical Review E* 69.4 (2004), p. 045104.

[18] N. L. Dehghani, S. Zamanian, and A. Shafieezadeh. "Adaptive network reliability analysis: Methodology and applications to power grid". In: *Reliability Engineering & System Safety* 216 (2021), p. 107973.

[19] T. Elperin, I. Gertsbakh, and M. Lomonosov. "An evolution model for Monte Carlo estimation of equilibrium network renewal parameters". In: *Probability in the Engineering and Informational Sciences* 6.4 (1992), pp. 457–469.

[20] T. Elperin, I. Gertsbakh, and M. Lomonosov. "Estimation of network reliability using graph evolution models". In: *IEEE Transactions on Reliability* 40.5 (1991), pp. 572–581.

[21] G. S. Fishman. "A Monte Carlo sampling plan for estimating network reliability". In: *Operations Research* 34.4 (1986), pp. 581–594.

[22] G. S. Fishman. "Monte Carlo estimation of the maximal flow distribution with discrete stochastic arc capacity levels". In: *Naval Research Logistics (NRL)* 36.6 (1989), pp. 829–849.

[23] S. Geyer, I. Papaioannou, and D. Straub. "Cross-entropy-based importance sampling using Gaussian densities revisited". In: *Structural Safety* 76 (2019), pp. 15–27.

[24] J. J. Grainger. *Power System Analysis*. McGraw-Hill, 1999.

[25] G. Hardy, C. Lucet, and N. Limnios. "K-terminal network reliability measures with binary decision diagrams". In: *IEEE Transactions on Reliability* 56.3 (2007), pp. 506–515.

[26] K.-P. Hui, N. Bean, M. Kraetzl, and D. Kroese. "Network reliability estimation using the tree cut and merge algorithm with importance sampling". In: *Proceedings of the 4th International Workshop on Design of Reliable Communication Networks*. IEEE. 2003, pp. 254–262.

[27] K.-P. Hui, N. Bean, M. Kraetzl, and D. P. Kroese. "The cross-entropy method for network reliability estimation". In: *Annals of Operations Research* 134.1 (2005), pp. 101–118.

[28] H. A. Jensen and D. J. Jerez. "A stochastic framework for reliability and sensitivity analysis of large scale water distribution networks". In: *Reliability Engineering & System Safety* 176 (2018), pp. 80–92.

[29] A. Kong. "A note on importance sampling using standardized weights". In: *University of Chicago, Dept. of Statistics, Tech. Rep* 348 (1992).

[30] D. P. Kroese, T. Taimre, and Z. I. Botev. *Handbook of Monte Carlo Methods*. Vol. 706. John Wiley & Sons, 2013.

[31] H. Kumamoto, K. Tanaka, K. Inoue, and E. J. Henley. "Dagger-sampling Monte Carlo for system unavailability evaluation". In: *IEEE Transactions on Reliability* 29.2 (1980), pp. 122–125.

[32] P. L'Ecuyer. "Efficiency improvement and variance reduction". In: *Proceedings of Winter Simulation Conference*. IEEE. 1994, pp. 122–132.

[33] J. Li and J. He. "A recursive decomposition algorithm for network seismic reliability evaluation". In: *Earthquake Engineering & Structural Dynamics* 31.8 (2002), pp. 1525–1539.

[34] J. Li and W. Liu. *Lifeline Engineering Systems: Network Reliability Analysis and Aseismic Design*. Springer Nature, 2021.

[35] H. O. Madsen, S. Krenk, and N. C. Lind. *Methods of Structural Safety*. Courier Corporation, 2006.

[36] H. Miao, W. Liu, and J. Li. "Seismic reliability analysis of water distribution networks on the basis of the probability density evolution method". In: *Structural Safety* 86 (2020), p. 101960.

[37] M. Moustapha and B. Sudret. "Learning non-stationary and discontinuous functions using clustering, classification and Gaussian process modelling". In: *arXiv preprint arXiv:2211.16909* (2022).

[38] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[39] A. B. Owen. *Monte Carlo Theory, Methods and Examples*. Standford, 2013.

[40] I. Papaioannou, S. Geyer, and D. Straub. "Improved cross-entropy-based importance sampling with a flexible mixture model". In: *Reliability Engineering & System Safety* 191 (2019), p. 106564.

[41] R. Paredes, L. Dueñas-Osorio, K. S. Meel, and M. Y. Vardi. "Principled network reliability approximation: A counting-based approach". In: *Reliability Engineering & System Safety* 191 (2019), p. 106472.

[42] J. E. Ramirez-Marquez and D. W. Coit. "A Monte Carlo simulation approach for approximating multi-state two-terminal reliability". In: *Reliability Engineering & System Safety* 87.2 (2005), pp. 253–264.

[43] H. Rosero-Velásquez and D. Straub. "Representative Natural Hazard Scenarios for Risk Assessment of Spatially Distributed Infrastructure Systems". In: *The 29th European Safety and Reliability Conference*. 2019, pp. 1–7.

[44] G. Rubino and B. Tuffin. *Rare Event Simulation Using Monte Carlo Methods*. John Wiley & Sons, 2009.

[45] R. Y. Rubinstein. "Optimization of computer simulation models with rare events". In: *European Journal of Operational Research* 99.1 (1997), pp. 89–112.

[46] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, 2016.

[47] A. Scherb, L. Garrè, and D. Straub. "Reliability and component importance in networks subject to spatially distributed hazards followed by cascading failures". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 3.2 (2017).

[48] F. Uribe, I. Papaioannou, Y. M. Marzouk, and D. Straub. "Cross-entropy-based importance sampling with failure-informed dimension reduction for rare event simulation". In: *SIAM/ASA Journal on Uncertainty Quantification* 9.2 (2021), pp. 818–847.

[49] Z. Wang and J. Song. "Cross-entropy-based adaptive importance sampling using von Mises-Fisher mixture for high dimensional reliability analysis". In: *Structural Safety* 59 (2016), pp. 42–52.

[50] E. Zio. *Monte Carlo Simulation: The Method*. Springer, 2013.

[51] E. Zio and N. Pedroni. "Reliability analysis of discrete multi-state systems by means of subset simulation". In: *Proceedings of the 17th ESREL Conference*. 2008, pp. 22–25.

[52] K. M. Zuev, S. Wu, and J. L. Beck. "General network reliability problem and its efficient solution by subset simulation". In: *Probabilistic Engineering Mechanics* 40 (2015), pp. 25–35.

Chapter 6

Bayesian improved cross entropy method with categorical mixture models

Original Publication

J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method with categorical mixture models". (Under review).

Abstract

We employ the Bayesian improved cross entropy (BiCE) method for rare event estimation in static networks and choose the categorical mixture as the parametric family to capture the dependence among network components. At each iteration of the BiCE method, the mixture parameters are updated through the weighted maximum a posteriori (MAP) estimate, which mitigates the overfitting issue of the standard improved cross entropy (iCE) method through a novel balanced prior, and we propose a generalized version of the expectation-maximization (EM) algorithm to approximate this weighted MAP estimate. The resulting importance sampling distribution is proved to be unbiased. For choosing a proper number of components K in the mixture, we compute the Bayesian information criterion (BIC) of each candidate K as a by-product of the generalized EM algorithm. The performance of the proposed method is investigated through a simple illustration, a benchmark study, and a practical application. In all these numerical examples, the BiCE method results in an efficient and accurate estimator that significantly outperforms the standard iCE method and the BiCE method with the independent categorical distribution.

6.1 Introduction

In February 2021, three heavy winter storms swept over Texas and triggered one of the worst energy network failures in Texas state history, which soon led to a severe power, food, and water shortage. A conservative estimate of the property damage is over 195 billion US dollars and more than 246 (estimated) people died during this event. These devastating consequences highlight the need for understanding and managing the reliability of infrastructure networks. This requires an effective means for quantifying the probability of survival or, conversely, the probability of failure of network systems.

In this context, the network is often simplified as a graph, whose edges or/and nodes are subjected to random failure. The network's performance is therefore a random variable and the probability that the network cannot deliver a certain level of performance is referred to as the failure probability p_f. Mathematically, p_f is defined through a performance function, $g(\cdot)$, which gives the safety margin of the network performance, and through a probabilistic input, $p_{\boldsymbol{X}}(\cdot)$, that quantifies the uncertainty of the system state $\boldsymbol{X} \triangleq [X_1, ...X_d, ..., X_D]^T$. X_d represents the state of the d-th component of the network, either edge or node, and D is the total number of components. In particular, p_f reads

$$p_f \triangleq \Pr\{g(\boldsymbol{X}) \leqslant 0\} = \sum_{\boldsymbol{x} \in \Omega_{\boldsymbol{X}}} \mathbb{I}\{g(\boldsymbol{X}) \leqslant 0\} p_{\boldsymbol{X}}(\boldsymbol{x}), \tag{6.1}$$

where $\Omega_{\boldsymbol{X}}$ is the sample space of $\boldsymbol{X}$, and $\mathbb{I}\{\cdot\}$ represents the indicator function. Note that $\boldsymbol{X}$ is often discrete in the context of network reliability assessment. Hence, in Eq. (6.1) the failure probability p_f is written as a summation of the input distribution $p_{\boldsymbol{X}}(\cdot)$ over the failure domain $F \triangleq \{\boldsymbol{x} \in \Omega_{\boldsymbol{X}} : g(\boldsymbol{x}) \leqslant 0\}$.

The static(or time-independent) performance of networks can often be measured by either connectivity or 'flow' [66]. For computer and communication networks, the connection among different parts of the network is of major concern, resulting in three different types of connectivity-based problems, namely the two terminals, K terminals, and all terminals connectivity problems [2], while for road networks and food supply chains, one is primarily interested in the 'flow' that a network can deliver, e.g., the maximum flow that can be transported from A to B. These flow-based problems involve multi-state (even continuous) components or/and network performance and can often be regarded as an extension of the connectivity-based problems [40].

In Table 6.1, we summarize three state-of-art methods for solving connectivity/flow-based problems, where CB is short for the counting-based method [19, 46], SSD is for the state-space decomposition [18, 1, 37, 38, 45], and CP is for creation process embedded methods [20, 28, 43, 58, 5, 6, 9, 10]. Other widely used methods include, sum of disjoint products [3], binary decision diagram [27], matrix-based system reliability method [31, 55], and various minimal-cutsets/pathsets-based methods, e.g., [47, 39, 8, 65].

For power grids and water supply systems, the 'flow' is often driven by the physical law (e.g. Kirchhoff's law for power flow) and operation strategies, and the network is not necessarily coherent. Hence, approaches built on the coherency assumption are not directly applicable. A set of methods have been proposed to solve such problems, among which sampling-based methods feature prominently. These include crude Monte Carlo simulation (MCS) [22, 61], subset simulation [62, 64, 30, 12, 66], adaptive importance sampling (IS) [32, 34, 12, 13], and active learning methods [7, 17].

Table 6.1: Comparison of different methods for connectivity-based problems

	CB	SSD	CP
introduction	[1,2]	[3-7]	[8-13]
not suitable for	small comp. failure prob.	large scale network	costly $g(\cdot)$
multi-state extension	unknown	possible	possible
coherent system	needed	needed	needed
error estimate	user-specific	reliability bound	relative error

We mainly focus on the static rare event estimation for network performance in this paper, and therefore, methods for time-dependent network reliability estimation such as the probability density evolution method (PDEM) [36] and modern stochastic process methods [40] are not included here.

Recently, the authors employed the improved cross entropy method (iCE) for solving network reliability problems and introduced a Bayesian approach to circumvent the overfitting issue of the standard iCE. The proposed method is termed Bayesian iCE (BiCE) [13]. Therein, the parametric model for approximating the optimal IS distribution is an independent categorical distribution and hence does not account for the dependence among components in the optimal IS distribution. This motivates the idea of employing a more flexible categorical mixture as the parametric model within the BiCE method. This parametric model can be updated at each iteration of the BiCE method by the generalized EM algorithm, which is introduced in this paper to approximate the maximum a posteriori (MAP) estimate of the mixture parameters given weighed samples. Note that the EM algorithm for estimating the MAP of a mixture model is well known [42]; herein we develop a modified version that accounts for the sample weights. The major contribution of this paper is to combine this generalized EM algorithm with the BiCE method for handling a more flexible mixture parametric family. We find that the proposed method, termed BiCE-CM, clearly outperforms the BiCE method with a single independent categorical distribution and provides better results than the standard iCE method. The key ingredient of the proposed method is a balanced Dirichlet prior that does not dominate but can still correct the potentially overfitted weighted MLE in the iCE. A number of components K in the categorical mixture is chosen adaptively through the Bayesian information criterion (BIC).

The paper is organized as follows: In Sec. 2, we summarize the basic ideas of iCE, followed by a brief introduction of the categorical mixture model and its approximated inference techniques in Sec. 3. The BiCE method with a categorical mixture parametric family (BiCE-CM) is introduced in Sec. 4. The efficiency and accuracy of the proposed method are demonstrated by a set of numerical examples in Sec. 5.

6.2 Cross-entropy-based importance sampling

In this section, we give a brief introduction to CE-based IS [16]. The basic idea is to choose the IS distribution from a predefined parametric family $h(\cdot;\boldsymbol{v})$ that best resembles the optimal IS

distribution

$$p^*_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{p_{\boldsymbol{X}}(\boldsymbol{x})\mathbb{I}\{g(\boldsymbol{x}) \leq 0\}}{p_f} = p_{\boldsymbol{X}}(\boldsymbol{x}|F). \tag{6.2}$$

The similarity between $p^*_{\boldsymbol{X}}(\cdot)$ and $h(\cdot;\boldsymbol{v})$ is measured by the Kullback–Leibler (KL) divergence that is defined as follows:

$$\begin{aligned} D(p^*_{\boldsymbol{X}}(\cdot), h(\cdot;\boldsymbol{v})) &= \mathbb{E}_{p^*_{\boldsymbol{X}}}\left[\ln\left(\frac{p^*_{\boldsymbol{X}}(\boldsymbol{X})}{h(\boldsymbol{X};\boldsymbol{v})}\right)\right] \\ &= \mathbb{E}_{p^*_{\boldsymbol{X}}}[\ln(p^*_{\boldsymbol{X}}(\boldsymbol{X}))] - \mathbb{E}_{p^*_{\boldsymbol{X}}}[\ln(h(\boldsymbol{X};\boldsymbol{v}))]. \end{aligned} \tag{6.3}$$

In other words, the CE method determines the optimal parameter vector $\boldsymbol{v}^*$ in $h(\cdot;\boldsymbol{v})$ through minimizing the KL divergence in Eq. (6.3), i.e., through solving

$$\begin{aligned} \boldsymbol{v}^* &= \arg\min_{\boldsymbol{v}\in\mathcal{V}} D(p^*_{\boldsymbol{X}}(\cdot), h(\cdot;\boldsymbol{v})) \\ &= \arg\min_{\boldsymbol{v}\in\mathcal{V}} -\mathbb{E}_{p^*_{\boldsymbol{X}}}[\ln(h(\boldsymbol{X};\boldsymbol{v}))] \\ &= \arg\max_{\boldsymbol{v}\in\mathcal{V}} \mathbb{E}_{p_{\boldsymbol{X}}}[\mathbb{I}\{g(\boldsymbol{X}) \leq 0\}\ln(h(\boldsymbol{X};\boldsymbol{v}))]. \end{aligned} \tag{6.4}$$

The problem in Eq. (6.4) cannot be solved in closed form due to the indicator function inside the expectation, so instead we estimate $\boldsymbol{v}^*$ through optimizing an alternative objective function that substitutes the expectation in Eq. (6.4) with an IS estimator. That is, we solve

$$\widehat{\boldsymbol{v}} = \arg\max_{\boldsymbol{v}\in\mathcal{V}} \frac{1}{N}\sum_{i=1}^{N} \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_i)\mathbb{I}\{g(\boldsymbol{x}_i) \leq 0\}}{p_{ref}(\boldsymbol{x}_i)} \ln(h(\boldsymbol{x}_i;\boldsymbol{v})), \qquad \boldsymbol{x}_i \sim p_{ref}(\cdot). \tag{6.5}$$

$\{\boldsymbol{x}_i\}_{i=1}^N$ are samples from $p_{ref}(\cdot)$, the IS distribution for estimating the expectation in Eq. (6.4), which is also known as the reference distribution [16]. Note that $\widehat{\boldsymbol{v}}$ can be interpreted as the weighted MLE of the parametric family with weights $\{w_i \propto \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_i)\mathbb{I}\{g(\boldsymbol{x}_i)\leq 0\}}{p_{ref}(\boldsymbol{x}_i)}\}_{i=1}^N$ [26, 13].

As discussed in [15, 13], one should distinguish the sub-optimal IS distribution $h(\cdot;\boldsymbol{v}^*)$ from the chosen IS distribution $h(\cdot;\widehat{\boldsymbol{v}})$ in the CE method. $h(\cdot;\boldsymbol{v}^*)$ is conditional on the predefined parametric family while $h(\cdot;\widehat{\boldsymbol{v}})$ additionally depends on the CE procedure, in particular, the choice of the reference distribution $p_{ref}(\cdot)$ and the number of samples. An appropriate reference distribution leads to an IS distribution $h(\boldsymbol{x};\widehat{\boldsymbol{v}})$ close to $h(\boldsymbol{x};\boldsymbol{v}^*)$, which is the optimal choice within the given parametric family.

For rare event estimation, the reference distribution is chosen in an adaptive way. Let $p^{(t)}_{\boldsymbol{X}}(\cdot), t = 1, ..., T$ denote a sequence of intermediate target distributions that gradually approach the optimal IS distribution $p^*_{\boldsymbol{X}}(\cdot)$. The CE optimization problem is then solved iteratively for finding a good approximation to each t-th $p^{(t)}_{\boldsymbol{X}}(\cdot)$, and this results in a sequence of CE parameter vectors $\{\widehat{\boldsymbol{v}}^{(t)}, t = 1, ..., T\}$ and distributions $\{h(\cdot;\widehat{\boldsymbol{v}}^{(t)}), t = 1, ..., T\}$. The distribution we obtain in the t-th iteration, i.e., $h(\cdot;\widehat{\boldsymbol{v}}^{(t)})$, is used as the reference distribution $p_{ref}(\cdot)$ for the CE procedure in iteration $t+1$. In this way, one takes $h(\cdot;\widehat{\boldsymbol{v}}^{(T-1)})$ as the reference distribution for Eq. (6.5), and $h(\cdot;\widehat{\boldsymbol{v}}^{(T)})$ as the final IS distribution. For the first iteration, the input distribution $p_{\boldsymbol{X}}(\cdot)$ is used as the reference distribution.

There are many different ways of designing the intermediate target distributions [16, 44, 57]. For instance, in the iCE method [44], the intermediate target distribution reads

$$p_{\boldsymbol{X}}^{(t)}(\boldsymbol{x}) \triangleq \frac{1}{Z^{(t)}} p_{\boldsymbol{X}}(\boldsymbol{x}) \Phi\left(-\frac{g(\boldsymbol{x})}{\sigma^{(t)}}\right), t = 1, ..., T \tag{6.6}$$

where $Z^{(t)}$ is the normalizing constant and Φ is the cumulative distribution function (CDF) of the standard normal distribution. The distribution sequence is driven by the parameter $\sigma^{(t)} > 0$, and gradually approaches the optimal IS distribution with decreasing $\sigma^{(t)}$. The CE optimization problem for Eq. (6.6) reads

$$\boldsymbol{v}^{(t,*)} = \arg\max_{\boldsymbol{v} \in \mathcal{V}} \mathbb{E}_{p_{\boldsymbol{X}}}[\Phi(-g(\boldsymbol{X})/\sigma^{(t)}) \ln(h(\boldsymbol{X}; \boldsymbol{v}))], \tag{6.7}$$

and the sample counterpart of Eq. (6.7) can be written as

$$\widehat{\boldsymbol{v}}^{(t)} = \arg\max_{\boldsymbol{v} \in \mathcal{V}} \frac{1}{N} \sum_{i=1}^{N} W(\boldsymbol{x}_i) \ln(h(\boldsymbol{x}_i; \boldsymbol{v})), \boldsymbol{x}_i \sim h(\cdot; \widehat{\boldsymbol{v}}^{(t-1)}) \tag{6.8}$$

$$W(\boldsymbol{x}_i) \triangleq \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_i)\Phi(-g(\boldsymbol{x}_i)/\sigma^{(t)})}{h(\boldsymbol{x}_i; \widehat{\boldsymbol{v}}^{(t-1)})}. \tag{6.9}$$

Note that $\widehat{\boldsymbol{v}}^{(t)}$ is the weighted maximum likelihood estimation (MLE) of $\boldsymbol{v}^{(t,*)}$, and for a properly reparameterized exponential family, $\widehat{\boldsymbol{v}}^{(t)}$ is also the self-normalized IS estimator of $\boldsymbol{v}^{(t,*)}$ [13]. The accuracy of $\widehat{\boldsymbol{v}}^{(t)}$ can be measured by the effective sample size (ESS), which is defined as the equivalent sample size required by MCS with the current target distribution to achieve the same variance as the self-normalized IS. The ESS of $\widehat{\boldsymbol{v}}^{(t)}$ in Eq. (6.8) can be approximated by [33]

$$ESS \approx \frac{N}{1 + \widehat{\delta}^2(\{W(\boldsymbol{x}_i)\}_{i=1}^N)}, \quad \boldsymbol{x}_i \sim h(\cdot; \widehat{\boldsymbol{v}}^{(t-1)}) \tag{6.10}$$

where $\widehat{\delta}(\{W(\boldsymbol{x}_i)\}_{i=1}^N)$ represents the sample coefficient of variation (c.o.v.) of the weights vector $\{W(\boldsymbol{x}_i)\}_{i=1}^N$. Although the categorical mixture employed in this paper does not belong to the exponential family, we still expect that a large ESS will generally lead to a more accurate $\widehat{\boldsymbol{v}}^{(t)}$.

Given the reference distribution $h(\boldsymbol{x}_i; \widehat{\boldsymbol{v}}^{(t-1)})$, the iCE method fixes N and changes $\sigma^{(t)}$ for achieving a constant ESS, and hence an accurate $\widehat{\boldsymbol{v}}^{(t)}$. Specifically, the intermediate target distribution in the iCE method is adapted at each t-th iteration by solving

$$\sigma^{(t)} = \arg\min_{\sigma \in (0, \sigma^{(t-1)})} |\widehat{\delta}\left(\{W(\boldsymbol{x}_i; \sigma)\}_{i=1}^N\right) - \delta_{tar}|, \qquad \boldsymbol{x}_i \sim h(\cdot; \widehat{\boldsymbol{v}}^{(t-1)}), \tag{6.11}$$

where $\widehat{\delta}(\cdot)$ represents the sample c.o.v. of a vector and δ_{tar} is the hyperparameter that influences the convergence rate of the intermediate target distributions. A common choice is $\delta_{tar} = 1.5$. The above procedure is iterated until

$$\widehat{\delta}\left(\left\{\frac{\mathbb{I}\{g(\boldsymbol{x}_i) \leq 0\}}{\Phi(-g(\boldsymbol{x}_i)/\sigma^{(t)})}\right\}_{i=1}^N\right) \leq \delta_\epsilon, \qquad \boldsymbol{x}_i \sim h(\cdot; \widehat{\boldsymbol{v}}^{(t)}). \tag{6.12}$$

where δ_ϵ is another hyperparameter and is often chosen to be the same as δ_{tar} [44].

It should be stressed that the standard iCE method may suffer from overfitting when the sample size is small. To mitigate this issue, the BiCE method [13] substitutes the weighted MLE with its

Bayesian counterpart; therein the posterior predictive distribution is employed to update a single categorical parametric model in the context of network reliability assessment. In addition, the BiCE method employs an alternative weight function for solving $\sigma^{(t)}$ through Eq. (6.11), which is defined as

$$W^{(alt)}(\boldsymbol{x};\sigma) \triangleq \frac{\Phi(-g(\boldsymbol{x})/\sigma)}{\Phi(-g(\boldsymbol{x})/\sigma^{(t-1)})}. \tag{6.13}$$

For a more detailed discussion and theoretical justification of Eq. (6.13), we refer to [14] and [13].

In this paper, we consider a more flexible parametric model, the categorical mixture, in the BiCE method. Before introducing the proposed CE approach, we first give an introduction to the categorical mixture model and its associated inference techniques in the following section.

6.3 The categorical mixture model

The categorical mixture model can be defined as:

$$h_{cm}(\boldsymbol{x};\boldsymbol{\eta}) = \sum_{k=1}^{K} \alpha_k h_c(\boldsymbol{x};\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \alpha_k \prod_{d=1}^{D} \prod_{j=1}^{n_d} \theta_{k,d,j}^{\mathbb{I}\{x_d = s_{d,j}\}}. \tag{6.14}$$

The probability distribution $h_{cm}(\cdot;\boldsymbol{\eta})$ is modelled as a linear combination of K independent categorical components, denoted here as $h_c(\cdot;\boldsymbol{\theta})$. In this paper, $h_c(\cdot;\boldsymbol{\theta})$ denotes the independent categorical distribution with parameters $\boldsymbol{\theta}$. Specifically, in the k-th mixture component, the probability that the d-th component X_d takes the j-th state $s_{d,j}$ is $\theta_{k,d,j}$, where $k = 1, ..., K; d = 1, ..., D; j = 1, ..., n_d$. D and n_d denote the number of input random variables X_d and the number of states for each X_d. $\alpha_k, k = 1, ...K$, are the non-negative mixture weights that sum to one. All model parameters are collected in the vector $\boldsymbol{\eta}$, i.e., $\boldsymbol{\eta} \triangleq \{\alpha_k, \boldsymbol{\theta_k})\}_{k=1}^{K}$.

The mixture model described in Eq. (6.14) is invariant with respect to the permutation of the component labels. As a result, the parameter estimation is unidentifiable [24]. Additionally, Eq. (6.14) remains invariant also (1) when adding a mixture component with zero weight, or (2) when replicating any of the mixture components and splitting the associated weight [24], which leads to a broader class of unidentifiability of the model parameters [53].

6.3.1 MLE of the categorical mixture and EM algorithm

Suppose we want to fit a categorical mixture described in Eq. (6.14) with N samples, $\mathcal{X} \triangleq \{\boldsymbol{x}_i\}_{i=1}^{N}$, and consider the case where the number of mixture components is known to be K. The most common approach is through MLE. The log-likelihood is

$$\ln \mathcal{L}(\boldsymbol{\eta};\mathcal{X}) \triangleq \ln \left(\prod_{i=1}^{N} h_{cm}(\boldsymbol{x}_i;\boldsymbol{\eta}) \right) = \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} \alpha_k h_c(\boldsymbol{x}_i;\boldsymbol{\theta}_k) \right). \tag{6.15}$$

The MLE for the categorical mixture cannot be obtained in closed form. If one observes the allocation variable z_i for each i-th sample $\boldsymbol{x}_i$, the log-likelihood function in Eq. (6.15) takes the following form:

$$\ln \mathcal{L}^{(c)}(\boldsymbol{\eta}; \mathcal{X}) = \sum_{i=1}^{N} \ln \left(\alpha_{z_i} h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_{z_i}) \right) = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \ln \left(\alpha_k h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_k) \right). \tag{6.16}$$

The allocation variable z_i specifies which mixture component generates $\boldsymbol{x}_i$, and $\mathcal{C}_k \triangleq \{i : i = 1, ..., N, z_i = k\}$ collects the indexes of all the samples generated by the k-th component of the mixture. Eq. (6.16) is often termed the complete data log-likelihood in the context of MLE to differentiate it from the log-likelihood in Eq. (6.15). Maximizing Eq. (6.16), is equivalent to fitting a categorical distribution $h_c(\cdot; \boldsymbol{\theta}_k)$ for each $\mathcal{C}_k$ and letting the associated weight α_k be proportional to $|\mathcal{C}_k|$, the number of samples in $\mathcal{C}_k$. Note that the closed-form solution to the MLE is well known for the single categorical distribution.

However, the allocation variables $\{z_i\}_{i=1}^N$ are not observed; they are latent variables. One approach is to estimate the latent variables through a clustering algorithm. However, clustering of categorical data is usually not straightforward, especially in the high-dimensional sample space.

For finding a mode of the log-likelihood function shown in Eq. (6.15), one usually resorts to the EM algorithm, which iteratively updates and optimizes the so-called Q function, an auxiliary function that computes the expectation of the complete data log-likelihood in Eq. (6.16). That is,

$$\begin{aligned} Q(\boldsymbol{\eta}; \{p_{Z_i}(\cdot)\}_{i=1}^N) &= \sum_{i=1}^{N} \mathbb{E}_{Z_i \sim p_{Z_i}(\cdot)} \left[\ln \left(\alpha_{Z_i} h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_{Z_i}) \right) \right] \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} p_{Z_i}(k) \ln \left(\alpha_k h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_k) \right), \end{aligned} \tag{6.17}$$

where $p_{Z_i}(\cdot)$ is a customary distribution for the i-th allocation variable Z_i. $p_{Z_i}(k)$ represents the probability that the i-th sample is generated by the k-th component of the mixture. Note that $p_{Z_i}(\cdot)$ can be an arbitrary distribution without necessarily being related to $\boldsymbol{\eta}$. According to Jensen's inequality, the log-likelihood function $\ln \mathcal{L}(\boldsymbol{\eta}; \mathcal{X})$ in Eq. (6.15) is bounded from below by the Q function plus a constant [42]. That is

$$\begin{aligned} \ln \mathcal{L}(\boldsymbol{\eta}; \mathcal{X}) &= \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} p_{Z_i}(k) \frac{\alpha_k h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_k)}{p_{Z_i}(k)} \right) \\ &\geqslant \sum_{i=1}^{N} \left(\sum_{k=1}^{K} p_{Z_i}(k) \ln \frac{\alpha_k h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_k)}{p_{Z_i}(k)} \right) \\ &= Q(\boldsymbol{\eta}; \{p_{Z_i}(\cdot)\}_{i=1}^N) + \sum_{i=1}^{N} \mathbb{H}(p_{Z_i}(\cdot)). \end{aligned} \tag{6.18}$$

$\mathbb{H}(p_{Z_i}(\cdot)) \triangleq \sum_{k=1}^{K} -p_{Z_i}(k) \ln(p_{Z_i}(k)) \geqslant 0$ is the entropy of the distribution $p_{Z_i}(\cdot)$ and is a constant with respect to $\boldsymbol{\eta}$. The inequality (6.18) takes the equal sign if

$$p_{Z_i}(k) = \frac{\alpha_k h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_k)}{\sum_{k'=1}^{N} \alpha_{k'} h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_{k'})} \triangleq \gamma_{i,k}(\boldsymbol{\eta}) \tag{6.19}$$

holds for each $k = 1, ..., K$ and $i = 1, ..., N$. $[\gamma_{i,k}(\boldsymbol{\eta})]_{N \times K}$ is also termed the responsibility matrix in the literature [42].

Eq. (6.19) indicates that, for any given $\boldsymbol{\eta}$ denoted as $\boldsymbol{\eta}^{(cur)}$, one can choose $p_{Z_i}(\cdot) = \gamma_{i,\cdot}(\boldsymbol{\eta}^{(cur)})$ for each Z_i, such that $\ln \mathcal{L}(\boldsymbol{\eta}^{(cur)}; \mathcal{X}) = Q\left(\boldsymbol{\eta}^{(cur)}; \boldsymbol{\eta}^{(cur)}\right) + C(\boldsymbol{\eta}^{(cur)})$, where $Q\left(\boldsymbol{\eta}^{(cur)}; \boldsymbol{\eta}^{(cur)}\right)$ is short for $Q\left(\boldsymbol{\eta}^{(cur)}; \{\gamma_{i,\cdot}(\boldsymbol{\eta}^{(cur)})\}_{i=1}^{N}\right)$ and $C(\boldsymbol{\eta}^{(cur)}) \triangleq \sum_{i=1}^{N} \mathbb{H}(\gamma_{i,\cdot}(\boldsymbol{\eta}^{(cur)}))$. This is also known as the expectation step (E step) of the EM algorithm, in which we compute the responsibility matrix $[\gamma_{i,k}(\boldsymbol{\eta}^{(cur)})]_{N\times K}$ via Eq. (6.19) and formulate the Q function.

In the next step, the maximization step or the M step for short, the EM algorithm fixes $p_{Z_i}(k) = \gamma_{i,k}(\boldsymbol{\eta}^{(cur)})$ for each i and k and maximizes the Q function over $\boldsymbol{\eta}$ to find a new $\boldsymbol{\eta}$ denoted as $\boldsymbol{\eta}^{(nxt)}$ whose Q function is larger than that of $\boldsymbol{\eta}^{(cur)}$, i.e., $Q\left(\boldsymbol{\eta}^{(nxt)}; \boldsymbol{\eta}^{(cur)}\right) \geqslant Q\left(\boldsymbol{\eta}^{(cur)}; \boldsymbol{\eta}^{(cur)}\right)$. Since the Q function (plus a constant) is a lower bound of the log-likelihood as shown in Inequality(6.18), the log-likelihood of $\boldsymbol{\eta}^{(nxt)}$ is also larger than that of $\boldsymbol{\eta}^{(cur)}$. In fact, we have $\ln \mathcal{L}(\boldsymbol{\eta}^{(nxt)}; \mathcal{X}) \geqslant Q\left(\boldsymbol{\eta}^{(nxt)}; \boldsymbol{\eta}^{(cur)}\right) + C(\boldsymbol{\eta}^{(cur)}) \geqslant Q\left(\boldsymbol{\eta}^{(cur)}; \boldsymbol{\eta}^{(cur)}\right) + C(\boldsymbol{\eta}^{(cur)}) = \ln \mathcal{L}(\boldsymbol{\eta}^{(cur)}; \mathcal{X})$. The point here is that optimizing the Q function is much easier than optimizing the log-likelihood function in Eq. (6.15). Specifically, the M step solves the following optimization problem:

$$\begin{aligned} \boldsymbol{\eta}^{(nxt)} &= \arg\max_{\boldsymbol{\eta}} Q(\boldsymbol{\eta}; \boldsymbol{\eta}^{(cur)}) \\ &= \arg\max_{\boldsymbol{\eta}} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{i,k}(\boldsymbol{\eta}^{(cur)}) \ln\left(\alpha_k h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_k)\right). \end{aligned} \tag{6.20}$$

For the categorical mixture shown in Eq. (6.14), the closed-form solution $\boldsymbol{\eta}^{(nxt)} = \{\alpha_k^{(nxt)}, \boldsymbol{\theta}_k^{(nxt)}\}_{k=1}^{K}$ to the optimization problem in Eq. (6.20) exists and is given by:

$$\alpha_k^{(nxt)} = \frac{\sum_{i=1}^{N} \gamma_{i,k}(\boldsymbol{\eta}^{(cur)})}{\sum_{k=1}^{K} \sum_{i=1}^{N} \gamma_{i,k}(\boldsymbol{\eta}^{(cur)})}, \tag{6.21}$$

$$\theta_{k,d,j}^{(nxt)} = \frac{\sum_{i=1}^{N} \gamma_{i,k}(\boldsymbol{\eta}^{(cur)}) \mathbb{I}\{x_{i,d} = s_{d,j}\}}{\sum_{i=1}^{N} \gamma_{i,k}(\boldsymbol{\eta}^{(cur)})}. \tag{6.22}$$

Note that if there is no sample equal to $s_{d,j}$, the probability assigned to that state, i.e., $\theta_{k,d,j}^{(t+1)}$, will become zero in each k-th mixture component, and this can lead to overfitting, as will be shown later in Sec. 4.1.

Through iterating the above two steps by setting $\boldsymbol{\eta}^{(cur)} = \boldsymbol{\eta}^{(nxt)}$, one ends up with a sequence of model parameters, $\boldsymbol{\eta}^{(0)}, \boldsymbol{\eta}^{(1)}, ..., \boldsymbol{\eta}^{(T)}$, that gradually improves the log-likelihood function. Although this does not strictly imply the convergence of the EM algorithm to a local maximum, usually this is the case.

$\boldsymbol{\eta}^{(0)}$ represents an initial guess of the model parameters. Given the sample set and the stopping criteria, the final estimate of the model parameters only relates to the choice of $\boldsymbol{\eta}^{(0)}$. A common strategy for getting an appropriate starting point is to first launch several short pilot runs of the EM algorithm, each with a different initialization, and then to choose the starting point for which the log-likelihood is the largest. It is noted that the EM algorithm can also start from the M step instead of the E step, which requires an initial guess of the $p_{Z_i}(\cdot)$ for each Z_i.

6.3.2 Bayesian inference

In the following, we adopt the Bayesian viewpoint to the inference of mixture models with K components and interpret the model parameters as random variables, $\boldsymbol{E}$, whose prior distribution is denoted as $p_{\boldsymbol{E}}(\boldsymbol{\eta})$. The posterior distribution of parameters $\boldsymbol{E}$ given $\mathcal{X}$ is given by Bayes' rule as

$$p_{\boldsymbol{E}|\mathcal{X}}(\boldsymbol{\eta}|\mathcal{X}) = \frac{\mathcal{L}(\boldsymbol{\eta}|\mathcal{X}) \cdot p_{\boldsymbol{E}}(\boldsymbol{\eta})}{p_{\mathcal{X}}(\mathcal{X})}. \tag{6.23}$$

The resulting predictive distribution reads

$$p_{\boldsymbol{X}|\mathcal{X}}(\boldsymbol{x}|\mathcal{X}) = \int_{\Omega_{\boldsymbol{E}}} h_{cm}(\boldsymbol{x}|\boldsymbol{\eta}) \cdot p_{\boldsymbol{E}|\mathcal{X}}(\boldsymbol{\eta}|\mathcal{X}) d\boldsymbol{\eta}, \tag{6.24}$$

which is an expectation of the mixture model with respect to the posterior distribution of model parameters. $\Omega_{\boldsymbol{E}}$ represents the sample space of $\boldsymbol{E}$. The posterior distribution, and hence also the predictive distribution, is not analytically tractable. Instead, the posterior distribution can be approximated through MCMC sampling,

$$p_{\boldsymbol{E}|\mathcal{X}}(\boldsymbol{\eta}|\mathcal{X}) \approx \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(\boldsymbol{\eta} - \boldsymbol{\eta}_i), \tag{6.25}$$

where $\delta(\cdot)$ is the Dirac delta function and $\{\boldsymbol{\eta}_i\}_{i=1}^{N_p}$ denotes the posterior samples. In this way, the predictive distribution is a mixture of mixtures consisting of a total of $N_p \cdot K$ mixture components. The computational cost of computing and sampling from this approximate predictive distribution is roughly N_p times the cost for a K-component mixture, and N_p is often large, say thousands. Therefore in this paper, we resort to a single point estimate of the model parameter, namely the MAP estimate $\widetilde{\boldsymbol{\eta}}$, for which the posterior distribution $p_{\boldsymbol{E}|\mathcal{X}}(\boldsymbol{\eta}|\mathcal{X})$ is maximized. Another benefit of using the MAP is that it can be obtained directly from the EM algorithm [42], which is significantly cheaper than running an MCMC algorithm.

The derivative of the EM algorithm for computing the MAP estimate follows the same lines as for the MLE, with a minor modification to account for the prior. Specifically, a log-prior distribution $\ln(p_{\boldsymbol{E}}(\boldsymbol{\eta}))$ is added to the original Q function in Eq. (6.17), and the EM algorithm proceeds iteratively with the following two steps: (1) E step: compute the distribution of the allocation variables $\boldsymbol{Z}$ through Eq. (6.19). (2) M step: update the model parameters through maximizing a modified Q function, i.e.,

$$\boldsymbol{\eta}^{(nxt)} = \arg\max_{\boldsymbol{\eta}} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{i,k}(\boldsymbol{\eta}^{(cur)}) \ln\left(\alpha_k h_c(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\right) + \ln(p_{\boldsymbol{E}}(\boldsymbol{\eta})). \tag{6.26}$$

In particular, for a conjugate prior distribution $p_{\boldsymbol{E}}(\boldsymbol{\eta})$, a closed-form updating scheme can be derived for the categorical mixture parameters.

6.3.3 Model selection and BIC

In this subsection, we discuss how to select the number of components K in the mixture model $h_{cm}(\cdot;\boldsymbol{\eta})$ using the information provided by the samples $\mathcal{X} \triangleq \{\boldsymbol{x}_i\}_{i=1}^{N}$. Let the initial pool of candidate models be $\{\mathcal{M}_K\}_{k=1}^{K_{max}}$ where $\mathcal{M}_K$ refers to a mixture of K independent categorical components

and K_{max} is a hyperparameter representing the maximum number of mixture components. From a Bayesian perspective, we favor the model $\mathcal{M}_{\widetilde{K}}$ with the highest posterior probability, or equivalently with the highest log-posterior. That is

$$\begin{aligned}\widetilde{K} &= \arg\max_{K} \ln p_{\mathcal{M}|\mathcal{X}}(\mathcal{M}_K|\mathcal{X}) \\ &= \arg\max_{K} \ln \mathcal{L}(\mathcal{M}_K|\mathcal{X}) + \ln p_{\mathcal{M}}(\mathcal{M}_K) \\ &= \arg\max_{K} \ln \left(\int_{\Omega_{\boldsymbol{E}}} \mathcal{L}(\boldsymbol{\eta}|\mathcal{X}, \mathcal{M}_K) p_{\boldsymbol{E}|\mathcal{M}}(\boldsymbol{\eta}|\mathcal{M}_K) d\boldsymbol{\eta} \right) + \ln p_{\mathcal{M}}(\mathcal{M}_K). \end{aligned} \tag{6.27}$$

Here, $p_{\mathcal{M}}(\mathcal{M}_K)$ represents the prior probability for each k-th candidate model, and it is often assumed to be uniformly distributed among all candidates. $\mathcal{L}(\mathcal{M}_K|\mathcal{X})$ denotes the integrated likelihood, or the marginal likelihood, and is the integral of the likelihood function $\mathcal{L}(\boldsymbol{\eta}|\mathcal{X}, \mathcal{M}_K)$ multiplied by the parameter prior distribution $p_{\boldsymbol{E}|\mathcal{M}}(\boldsymbol{\eta}|\mathcal{M}_K)$ over the whole sample space of the parameters $\Omega_{\boldsymbol{E}}$. Note that this is actually the normalizing constant of the posterior distribution of the parameters in $\mathcal{M}_K$, i.e., $p_{\boldsymbol{E}|\mathcal{X},\mathcal{M}}(\boldsymbol{\eta}|\mathcal{X}, \mathcal{M}_K)$.

Computing the integrated likelihood involves a high dimensional integration whose closed-form solution is not available. Nevertheless, it can be approximated through various sampling-based methods [25, 23, 41]. These methods often rely on computationally expensive MCMC algorithms and are limited to a small K, for example, up to 6 [24]. The Bayesian information criterion (BIC) serves as a crude but computationally cheap proxy of the log-posterior probability when $p_{\mathcal{M}}(\mathcal{M}_K) \propto 1$. BIC was first introduced by Schwarz [54] for asymptotically approximating the log-posterior probability of a linear model given observations $\mathcal{X}$ from a regular exponential family (see the definition in [54]); therein the BIC is defined as $\ln \mathcal{L}(\widehat{\boldsymbol{\eta}}|\mathcal{X}, \mathcal{M})) - \frac{\dim(\mathcal{M}) \ln(N)}{2}$, where $\ln \mathcal{L}(\widehat{\boldsymbol{\eta}}|\mathcal{X}, \mathcal{M})$ represents the mode of the log-likelihood function evaluated at the MLE point $\widehat{\boldsymbol{\eta}}$, and $\dim(\mathcal{M})$ denotes the number of free parameters in $\mathcal{M}$. Another commonly used definition is given by

$$\mathrm{BIC}(\mathcal{M}) \triangleq -2 \ln \mathcal{L}(\widehat{\boldsymbol{\eta}}|\mathcal{X}, \mathcal{M}) + \dim(\mathcal{M}) \ln(N). \tag{6.28}$$

Note that under the definition of Eq. (6.28), the model with the smallest BIC is favored.

The derivation of the BIC relies on the Laplace approximation to the likelihood function $\mathcal{L}(\boldsymbol{\eta}|\mathcal{X}, \mathcal{M})$, which does not apply to multi-modal posterior distributions, and thus BIC cannot be interpreted as a meaningful approximation to the log-posterior of a mixture model. In spite of this, BIC remains one of the state-of-art techniques for selecting the number of mixture components in practice[51, 56, 4, 24]. Additionally, BIC can be computed directly as a by-product of the EM algorithm without employing any computationally expensive MCMC algorithm. Therefore, throughout this paper, we adopt the BIC as the model selection technique.

6.4 Bayesian improved cross entropy method with the categorical mixture model

In this section, we introduce the Bayesian iCE method with the categorical mixture model for network reliability analysis. With slight abuse of notation, we omit the subscript for all prior and posterior distributions, and use, e.g., $p(\boldsymbol{\eta})$ to represent $p_{\boldsymbol{E}}(\boldsymbol{\eta})$.

6.4.1 Motivation

As mentioned in Sec. 6.2, the 'distance' between the optimal IS distribution and the suboptimal IS distribution is only related to the chosen parametric model. For a fixed parametric model, the 'distance' remains fixed assuming that the CE optimization problem is solved exactly. An inappropriate parametric model will lead to an IS estimator with large variance in the final level of CE-based methods. In particular, this can happen when approximating an optimal IS distribution that implies a strong dependence between component states with the independent categorical model.

To account for the dependence between the component states, one could use a dependent categorical distribution. However, it is not straightforward to choose an appropriate dependence structure that is both easy to sample from and convenient to update. Instead, we consider the mixture of independent categorical distributions as the parametric model. The flexibility of this mixture model enables capturing arbitrary dependencies between variables in the optimal IS distribution. In the CE-based IS, the parametric model is updated by maximizing a weighted log-likelihood function as shown in Eq. (6.8). Therefore, techniques for MLE can also be used in the CE-based methods with minor modifications to account for the weights. For instance, Geyer et.al., [26] used the EM algorithm for updating a Gaussian mixture model in the CE method. They found that the Gaussian mixture model performs consistently worse than the single Gaussian model especially when the sample size is small. The reason is that the EM algorithm tends to overfit the weighted samples and hence it is more sensitive to sample sets that misrepresent the target distribution. This can happen when the geometry/shape of the intermediate target distributions changes significantly in CE-based methods, which results in one or more modes of the target distribution being missing or cannot be sufficiently reflected by the weighted samples. The overfitting issue is even more severe for updating the categorical mixture in CE methods. If there is no sample falling into a certain category during the adaptive process, the probability assigned to this category will be zero for all mixture components, resulting in a potentially biased estimate of the final IS estimator. This is also known as the zero count problem in the context of MLE with categorical data [42]. A detailed discussion of the zero count problem for the CE method with the independent categorical parametric model can be found in [13].

6.4.2 Bayesian updating for cross-entropy-based methods

6.4.2.1 The basic idea

To circumvent the overfitting issue of the weighted MLE, we propose to use the Bayesian approach for updating the categorical mixture in the CE method. At each level, we approximate the weighted MAP of a K-component mixture, denoted as $\widetilde{\boldsymbol{\eta}}|\mathcal{M}_K$, through a generalized version of the EM algorithm that works with weighted samples. Here, we use 'approximate' to indicate that the algorithm is prone to get stuck in a local maximum, but this limitation can be alleviated by launching short pilot runs as mentioned in Subsec. 6.3.1. Model selection is performed for estimating the optimal number of components $\widetilde{K}$ in the categorical mixture, whereby the number of mixture components leading to the smallest BIC is selected. Next, we employ the $\widetilde{K}$-component categorical mixture with its parameters fixed at $\widetilde{\boldsymbol{\eta}}|\mathcal{M}_{\widetilde{K}}$ as the reference/sampling distribution at the $(t+1)$-th level in the CE method. We term the proposed method BiCE-CM.

6.4.2.2 The generalized EM algorithm

In this subsection, we introduce a generalized version of the EM algorithm and demonstrate its properties. To this end, we first attach a Dirichlet prior, which is the conjugate prior for categorical distributions, to each model parameter, i.e.,

$$\begin{aligned}\boldsymbol{\alpha} &\triangleq \{\alpha_k\}_{k=1}^K \sim \mathrm{Dir}(\cdot|\boldsymbol{a}) \\ \boldsymbol{\theta}_{k,d} &\triangleq \{\theta_{k,d,j}\}_{j=1}^{n_d} \sim \mathrm{Dir}(\cdot|\boldsymbol{b}_{k,d}) \\ p(\boldsymbol{\eta}|\mathcal{M}_K) &= \mathrm{Dir}(\boldsymbol{\alpha}|\boldsymbol{a}) \prod_{k=1}^K \prod_{d=1}^D \mathrm{Dir}(\boldsymbol{\theta}_{k,d}|\boldsymbol{b}_{k,d}),\end{aligned} \tag{6.29}$$

where $\boldsymbol{a} = (a_1, ..., a_k)$ and $\boldsymbol{b}_{k,d} = (b_{k,d,1}, ..., b_{k,d,n_d})$ are predefined concentration parameters. We obtain an MAP estimate of the model parameters $\boldsymbol{\eta}$ through maximizing the weighted log-posterior distribution $\ln\left(p^{(w)}(\boldsymbol{\eta}|\mathcal{X}, \mathcal{M}_K)\right)$, which reads:

$$\begin{aligned}\ln\left(p^{(w)}(\boldsymbol{\eta}|\mathcal{X}, \mathcal{M}_K)\right) &= \ln \mathcal{L}^{(w)}(\boldsymbol{\eta}|\mathcal{X}, \mathcal{M}_K)) + \ln(p(\boldsymbol{\eta}|\mathcal{M}_K)) \\ &= \sum_{i=1}^N w_i \ln\left(h_{cm}(\boldsymbol{x}_i|\boldsymbol{\eta})\right) + \ln(p(\boldsymbol{\eta}|\mathcal{M}_K)),\end{aligned} \tag{6.30}$$

where $\mathcal{L}^{(w)}(\boldsymbol{\eta}|\mathcal{X}, \mathcal{M}_K)$ is the weighted likelihood with $w_i \triangleq \frac{NW(\boldsymbol{x}_i)}{\sum_{j=1}^N W(\boldsymbol{x}_j)}$ representing the normalized weight of the i-th sample $\boldsymbol{x}_i$; herein, the weight function $W(\cdot)$ defined in Eq. (6.9) is normalized such that the sum of the weights is equal to N. Note that normalizing the weights $\{W(\boldsymbol{x}_i)\}_{i=1}^N$ does not change the solution to the original CE optimization problem in Eq. (6.8), i.e., $\widehat{\boldsymbol{v}}^{(t)}$, but can modify the relative strength between the log-prior and the weighted log-likelihood term in Eq. (6.30). As the sample size N increases, the log-prior term will be dominated by the weighted log-likelihood, and hence, the solution to Eq. (6.30) coincides with the results obtained from Eq. (6.8) in large sample settings. On the other hand, when the sample size is small/moderate, the prior term serves as a regularizer that penalizes the weighted log-likelihood. Different kinds of prior distributions or regularizers can be applied depending on the problems at hand, but a detailed investigation is left for future work. In this paper, we focus on the Dirichlet prior as shown in Eq. (6.29).

A generalized version of the EM algorithm is employed to maximize Eq. (6.30), which iteratively updates the following weighted Q function

$$\begin{aligned}Q^{(w)}(\boldsymbol{\eta}; \{p_{Z_i}(\cdot)\}_{i=1}^N) &\triangleq \sum_{i=1}^N w_i \mathbb{E}_{Z_i \sim p_{Z_i}(\cdot)}\left[\ln\left(\alpha_{Z_i} h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_{Z_i})\right)\right] + \ln(p(\boldsymbol{\eta}|\mathcal{M}_K)) \\ &= \sum_{i=1}^N w_i \sum_{k=1}^K p_{Z_i}(k)\left[\ln\left(\alpha_k h_c(\boldsymbol{x}_i; \boldsymbol{\theta}_k)\right] + \ln(p(\boldsymbol{\eta}|\mathcal{M}_K)\right).\end{aligned} \tag{6.31}$$

In the E step, we compute the responsibility matrix $[\gamma_{i,k}(\boldsymbol{\eta}^{(cur)})]_{N\times K}$ via Eq. (6.19) and formulate $Q^{(w)}(\boldsymbol{\eta}; \boldsymbol{\eta}^{(cur)}) \triangleq Q^{(w)}(\boldsymbol{\eta}; \{\gamma_{i,\cdot}(\boldsymbol{\eta}^{(cur)})\}_{i=1}^N)$; in the M step, we maximize $Q^{(w)}(\boldsymbol{\eta}; \boldsymbol{\eta}^{(cur)})$ over $\boldsymbol{\eta}$,

resulting in the following updating scheme for the categorical mixture:

$$\alpha_k^{(nxt)} = \frac{\sum_{i=1}^N w_i \gamma_{i,k}(\boldsymbol{\eta}^{(cur)}) + a_k - 1}{\sum_{k=1}^K \sum_{i=1}^N w_i \gamma_{i,k}(\boldsymbol{\eta}^{(cur)}) + \sum_{k=1}^K a_k - K}, \tag{6.32}$$

$$\theta_{k,d,j}^{(nxt)} = \frac{\sum_{i=1}^N w_i \gamma_{i,k}(\boldsymbol{\eta}^{(cur)}) \mathbb{I}\{x_{i,d} = s_{d,j}\} + b_{k,d,j} - 1}{\sum_{i=1}^N w_i \gamma_{i,k}(\boldsymbol{\eta}^{(cur)}) + \sum_{j=1}^{n_d} b_{k,d,j} - n_d}. \tag{6.33}$$

Similarly to the original EM algorithm, it holds that

$$\begin{aligned} \ln\left(p^{(w)}(\boldsymbol{\eta}^{(nxt)}|\mathcal{X}, \mathcal{M}_K)\right) &\geqslant Q^{(w)}(\boldsymbol{\eta}^{(nxt)}; \boldsymbol{\eta}^{(cur)}) + C^{(w)}(\boldsymbol{\eta}^{(cur)}) \\ &\geqslant Q^{(w)}(\boldsymbol{\eta}^{(cur)}; \boldsymbol{\eta}^{(cur)}) + C^{(w)}(\boldsymbol{\eta}^{(cur)}) = \ln\left(p^{(w)}(\boldsymbol{\eta}^{(cur)}|\mathcal{X}, \mathcal{M}_K)\right), \end{aligned} \tag{6.34}$$

where $C^{(w)}(\boldsymbol{\eta}^{(cur)}) \triangleq \sum_{i=1}^N w_i \mathbb{H}(\gamma_{i,\cdot}(\boldsymbol{\eta}^{(cur)}))$. We end up with a sequence of parameters $\boldsymbol{\eta}^{(0)}, ..., \boldsymbol{\eta}^{(T)}$ that converges to one of the modes (or saddle points) of the weighted log-posterior distribution, and $\boldsymbol{\eta}^{(T)}$ is regarded as an approximate weighted MAP, $\widetilde{\boldsymbol{\eta}}|\mathcal{M}_K$.

6.4.2.3 The weighted MAP mitigates the overfitting and is unbiased

$\boldsymbol{\eta}^{(T)}$ can be written as a linear combination of a data-dependent estimate $\boldsymbol{\eta}^{(T;\mathrm{D})}$, which exploits the current data, and a user-defined prior estimate $\boldsymbol{\eta}^{(T;\mathrm{pri})}$, which can be designed to explore a wider part of the sample space and thus is capable of finding potentially missing modes. Taking $\theta_{k,d,j}^{(T)}$ as an example, let $nxt = T, cur = T-1$ and rearrange Eq. (6.33) as follows:

$$\theta_{k,d,j}^{(T)} = \lambda_{k,d}(\boldsymbol{\eta}^{(T-1)}) \theta_{k,d,j}^{(nxt;\mathrm{D})} + (1 - \lambda_{k,d}(\boldsymbol{\eta}^{(T-1)})) \theta_{k,d,j}^{(\mathrm{pri})}. \tag{6.35}$$

where $\theta_{k,d,j}^{(T;\mathrm{D})} \triangleq \frac{\sum_{i=1}^N w_i \gamma_{i,k}(\boldsymbol{\eta}^{(T-1)}) \mathbb{I}\{x_{i,d} = s_{d,j}\}}{\sum_{i=1}^N w_i \gamma_{i,k}(\boldsymbol{\eta}^{(T-1)})}$, and $\theta_{k,d,j}^{(\mathrm{pri})} \triangleq \frac{b_{k,d,j} - 1}{\sum_{j=1}^{n_d} b_{k,d,j} - n_d}$. $\theta_{k,d,j}^{(T;\mathrm{D})}$ and $\theta_{k,d,j}^{(\mathrm{pri})}$ are combined via

$$\lambda_{k,d}(\boldsymbol{\eta}^{(T-1)}) \triangleq \frac{\sum_{i=1}^N w_i \gamma_{i,k}(\boldsymbol{\eta}^{(T-1)})}{\sum_{i=1}^N w_i \gamma_{i,k}(\boldsymbol{\eta}^{(T-1)}) + \sum_{j=1}^{n_d} b_{k,d,j} - n_d}, \tag{6.36}$$

which is a factor indicating the relative strength of the data with respect to the combined information from the data and prior. $\lambda_{k,d}(\boldsymbol{\eta}^{(T-1)})$ tunes the exploitation and exploration behaviour of $\theta_{k,d,j}^{(T)}$; the larger $\lambda_{k,d}(\boldsymbol{\eta}^{(T-1)})$ is, the more dominant is $\theta_{k,d,j}^{(T;\mathrm{D})}$ in Eq. (6.35). A similar interpretation also applies to $\alpha_k^{(T)}$. Moreover, if we set $b_{k,d,j} > 1$ for each k,d and j, $\theta_{k,d,j}^{(T)}$ is always positive even when no samples fall into the category $s_{d,j}$, i.e., the zero count issue is mitigated in small sample settings. As a result, the sample space of the reference distribution at each intermediate level will no longer shrink even with a small number of samples, which ensures an **unbiased** IS estimator at the final CE level.

6.4.2.4 Implementation details

6.4.2.4.1 Initialization

To initialize the generalized EM algorithm, we launch several short pilot runs, each from a random realization of the responsibility matrix $[\gamma_{i,k}^{(0)}]_{N \times K}$. The i-th row of the responsibility matrix is a K-

component vector generated uniformly and independently over the standard $(K-1)$-simplex, i.e., the vector follows the symmetric Dirichlet distribution $\mathrm{Dir}(\cdot|[1,...,1])$. The responsibility matrix that achieves the highest weighted log-posterior is chosen as the starting point from which we iteratively perform the M step and E step until convergence.

6.4.2.4.2 The prior distribution

For selecting an appropriate Dirichlet prior distribution in the BiCE-CM, we rearrange Eq. (6.36) as follows:

$$\sum_{j=1}^{n_d} b_{k,d,j} - n_d = \frac{\left(1-\lambda_{k,d}(\boldsymbol{\eta}^{T-1})\right)}{\lambda_{k,d}(\boldsymbol{\eta}^{T-1})} \cdot \sum_{i=1}^{N} w_i \gamma_{i,k}\left(\boldsymbol{\eta}^{T-1}\right). \tag{6.37}$$

For simplicity, let $\gamma_{i,k}\left(\boldsymbol{\eta}^{T-1}\right) = 1/K$ for each i and k, and assume a symmetric Dirichlet prior for each $\boldsymbol{\theta}_{k,d}$, i.e., $b_{k,d,j_1} = b_{k,d,j_2}$ for $1 \leqslant j_1 \neq j_2 \leqslant n_d$ and $1 \leqslant k \leqslant K, 1 \leqslant d \leqslant D$. $\boldsymbol{\theta}_{k,d}$ represents the PMF of X_d implied by the k-th component of the mixture. As a consequence, Eq. (6.37) can be written as

$$b_{k,d,j} = 1 + \frac{\left(1-\lambda_{k,d}(\boldsymbol{\eta}^{T-1})\right) \cdot \sum_{i=1}^{N} w_i}{\lambda_{k,d}(\boldsymbol{\eta}^{T-1}) \cdot K \cdot n_d}; \qquad j = 1,...,n_d. \tag{6.38}$$

We then replace $\frac{\left(1-\lambda_{k,d}(\boldsymbol{\eta}^{T-1})\right)}{\lambda_{k,d}(\boldsymbol{\eta}^{T-1})} \cdot \sum_{i=1}^{N} w_i$ by a constant C in Eq. (6.38), which gives

$$b_{k,d,j} = 1 + \frac{C}{K \cdot n_d}; \qquad j = 1,...,n_d \tag{6.39}$$

for each $\boldsymbol{\theta}_{k,d}$. We will compare different choices of C in the numerical examples. Note that $\sum_{i=1}^{N} w_i$ reflects the size (or strength) of the weighted data, and $\frac{C}{K}$ can be interpreted as the size (or strength) of the prior in each mixture component. As for the mixture weights $\boldsymbol{\alpha}$, we choose

$$a_k = 1 + \epsilon; \qquad k = 1,...,K, \tag{6.40}$$

where ϵ is typically set as a small value, e.g. 10^{-8}.

In fact, we penalize the weighted log-likelihood with the following log-prior term

$$\ln\left(p(\boldsymbol{\eta}|\mathcal{M}_k)\right) = \ln \mathrm{Dir}(\boldsymbol{\alpha}|\boldsymbol{a}) + \sum_{k=1}^{K}\sum_{d=1}^{D} \ln \mathrm{Dir}(\boldsymbol{\theta}_{k,d}|\boldsymbol{b}_{k,d}). \tag{6.41}$$

For symmetric Dirichlet distributions $\mathrm{Dir}(\boldsymbol{\alpha}|\boldsymbol{a})$ and $\mathrm{Dir}(\boldsymbol{\theta}_{k,d})$ defined in Eq. (6.39) and (6.40), the probability mode is attained when $\alpha_k = 1/K, k = 1,...,K$ and $\theta_{k,d,j} = 1/n_d, j = 1,...,n_d$. In other words, we favor a uniform vector for each $\boldsymbol{\theta}_{k,d}$, and a larger C implies a stronger preference. Note that by selecting a small ϵ, the penalization of non-uniform $\boldsymbol{\alpha}$ vanishes, so the redundant mixture components can be assigned a small weight.

6.4.2.4.3 Model selection or not

To discuss whether it is necessary to perform model selection, we consider two categorical mixtures $f_{m1}(\cdot|\boldsymbol{\eta}, \mathcal{M}_{K1})$ and $f_{m2}(\cdot|\boldsymbol{\eta}, \mathcal{M}_{K2})$. Let $K1 > K2$ and we refer to f_{m1}, f_{m2} as the larger mixture,

and the smaller mixture, respectively. Through, for example, adding $K1 - K2$ redundant mixture components, each of zero weight to f_{m2}, any distribution that can be represented by the smaller mixture f_{m2} can also be represented by the larger one f_{m1}. Therefore, the minimum KL divergence between the optimal IS distribution and the larger mixture will be less or equal to that of the smaller mixture, and if we can always find the optimal parameters $\boldsymbol{\eta}^*$ defined in Eq. (6.7), the BiCE-CM with a larger mixture will perform better or at least equally well than using a smaller mixture.

If the sample size approaches infinity, the distribution implied by either the weighted MLE $\widehat{\boldsymbol{\eta}}$ or the weighted MAP $\widetilde{\boldsymbol{\eta}}$ converges to the distribution implied by the optimal parameters $\boldsymbol{\eta}^*$, and if we can always find the weighted MLE or weighted MAP through the generalized EM algorithm, there is no need to perform model selection, since the larger the K, the closer the chosen IS distribution is to the optimal IS distribution, and thus the better the performance of the CE method.

In practical settings, the sample size is limited, and the weighted MLE $\widehat{\boldsymbol{\eta}}$ can be far away from the optimal parameter $\boldsymbol{\eta}^*$. Although by introducing the prior information, the overfitting issue of the weighted MLE is mitigated, there is still no guarantee that the distribution implied by the weighted MAP $\widetilde{\boldsymbol{\eta}}$ is close to that of $\boldsymbol{\eta}^*$. Even if the weighted MAP of a mixture can be found, it does not necessarily lead to a closer distribution to the optimal IS distribution than using the weighted MAP of a smaller mixture, especially when an inappropriate prior distribution is chosen, and hence, we cannot simply employ a large K.

Another major issue is that in practice the generalized EM algorithm almost always gets stuck at a local maximum and fails to identify the weighted MAP. Note that there are in total K^n terms (usually uni-modal) in the likelihood function. Although some of these terms can be merged, a large sample size n or number of mixture components K generally indicates a more complicated and jagged posterior surface, whereby our generalized EM optimizer is more likely to get stuck at a point far from optimal. In such cases, a higher effort is required to find a good local maximum, e.g., by launching more pilot runs or designing a special prior that eliminates some of the modes.

In summary, it is challenging to make a general decision on whether or not to perform the model selection, and we select the K with the highest posterior probability among a set of K_{max} candidates. The posterior probability can be roughly approximated by twice the negative BIC in Eq. (6.28). Although such an approximation suffers from major limitations, it remains one of the state-of-art techniques for selecting the number of components in a mixture model. For more details, we refer to Sec. 6.3.3.

6.4.2.4.4 The algorithm

The proposed generalized EM algorithm for inference of the categorical mixture is summarized in Algorithm 8.

Algorithm 8: The generalized EM algorithm

MainFunc:

Input: $\{\boldsymbol{x}_i, W_i \triangleq W(\boldsymbol{x}_i)\}_{i=1}^N$, C, ϵ, K, $\Omega_{\boldsymbol{X}} \triangleq \{s_{d,1}, ..., s_{d,n_d}\}_{d=1}^D$

% $\Omega_{\boldsymbol{X}}$ is the sample space of $\boldsymbol{X}$, $W(\cdot)$ is defined by Eq. (6.9)

$w_i \leftarrow N \cdot \frac{W_i}{\sum_{i=1}^N W_i}$ for each $i = 1, ..., N$ % normalizing the weights

$n_p \leftarrow 20$ % the number of the pilot runs

$l_p \leftarrow 20$ % the maximum iteration of the pilot run

$l_o \leftarrow 500$ % the maximum iteration of the official run

$\mathcal{LP}_{max} \leftarrow -\infty$ % the maximum weighted log-posterior of the pilot runs

$it \leftarrow 1$ % the counter for the pilot run

while $it \leqslant n_p$ **do**

  for $i = 1, ..., N$ **do**

    Generate $\{\gamma_{i,k}^{(0,it)}\}_{k=1}^K$ uniformly over the standard $(K-1)$ simplex

  $(\sim, \mathcal{LP}, \sim) =$ **Subroutine** $\left(\{\boldsymbol{x}_i, w_i\}_{i=1}^N, [\gamma_{i,k}^{(0,it)}]_{N\times K}, \Omega_{\boldsymbol{X}}, C, \epsilon, l_p\right)$

  if $\mathcal{LP} \geqslant \mathcal{LP}_{max}$ **then**

    $\gamma_{i,k}^{(0)} \leftarrow \gamma_{i,k}^{(0,it)}$ for each i and k, $\mathcal{LP}_{max} \leftarrow \mathcal{LP}$

  $it = it + 1$

$(\mathcal{LL}, \mathcal{LP}, \widetilde{\boldsymbol{\mu}}_K) =$ **Subroutine** $\left(\{\boldsymbol{x}_i, w_i\}_{i=1}^N, [\gamma_{i,k}^{(0)}]_{N\times K}, \Omega_{\boldsymbol{X}}, C, \epsilon, l_o\right)$

Compute BIC_K through Eq. (6.28)

Output: $\widetilde{\boldsymbol{\mu}}_K$, BIC_K

Subroutine:

Input: $\{\boldsymbol{x}_i, w_i\}_{i=1}^N, [\gamma_{i,k}]_{N\times K}, \Omega_{\boldsymbol{X}}, C, \epsilon, t_{max}$

$tol \leftarrow \frac{1}{10\cdot N}$, $r \leftarrow \infty$, $t \leftarrow 1$, $\mathcal{LP}^{(0)} \leftarrow 1$

while $r \geqslant tol$ *and* $t \leqslant t_{max}$ **do**

  M step: plug $\gamma_{i,k}$, w_i and $\Omega_{\boldsymbol{X}}$ into Eq. (6.32) and (6.33) and compute α_k and $\theta_{k,d,j}$ with a_k and $b_{k,d,j}$ defined in Eq. (6.40) and (6.39), respectively

  E step: update $\gamma_{i,k}$ through Eq. (6.19).

  compute the weighted log-likelihood $\mathcal{LL}^{(t)}$ and the weighted log-posterior $\mathcal{LP}^{(t)}$ via Eq. (6.30)

  $r \leftarrow \frac{|\mathcal{LP}^{(t)} - \mathcal{LP}^{(t-1)}|}{\mathcal{LP}^{(t-1)}}$, $t \leftarrow t + 1$

let $\widetilde{\boldsymbol{\mu}}$ collect all α_k and $\theta_{k,d,j}$

$\mathcal{LL} \leftarrow \mathcal{LL}^{(t-1)}$, $\mathcal{LP} \leftarrow \mathcal{LP}^{(t-1)}$

Output: $\mathcal{LL}$, $\mathcal{LP}$, $\widetilde{\boldsymbol{\mu}}$

6.4.3 Bayesian improved cross entropy method with the categorical mixture model

The BiCE method [13] substitutes the weighted MLE of model parameters in the original iCE method with a Bayesian counterpart. In [13], the posterior predictive distribution is derived for updating the independent categorical distribution. However, for the categorical mixture, a closed-form expression of the posterior predictive distribution does not exist, and we use the weighted

MAP estimator instead, which can be approximated through a generalized EM algorithm described in Subsec. 6.4.2. The proposed BiCE method with the categorical mixture model (BiCE-CM) is summarized in Algorithm 9.

Algorithm 9: Bayesian improved cross entropy method with the categorical mixture parametric family

Input: N, δ_{tar}, δ_ϵ, C, ϵ, the maximum number of mixture components K_{max}, performance function $g(\boldsymbol{x})$, input distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega_{\boldsymbol{X}}$

$t \leftarrow 1$, $t_{max} \leftarrow 50$, $\sigma_0 \leftarrow \infty$

$h(\boldsymbol{x}; \widetilde{\boldsymbol{\mu}}^{(t-1)}) \leftarrow p_{\boldsymbol{X}}(\boldsymbol{x})$

while *true* **do**

  Generate N samples $\{\boldsymbol{x}_k\}_{k=1}^N$ from $h(\boldsymbol{x}; \widetilde{\boldsymbol{\mu}}^{(t-1)})$ and calculate the corresponding performance $\{g(\boldsymbol{x}_k)\}_{k=1}^N$

  Compute the sample c.o.v. $\widehat{\delta}$ of $\left\{ \frac{\mathbb{I}\{g(\boldsymbol{x}_k) \leq 0\}}{\Phi(-g(\boldsymbol{x}_k)/\sigma^{(t-1)})} \right\}_{k=1}^N$

  if $t > t_{max}$ *or* $\widehat{\delta} \leq \delta_\epsilon$ **then**

    Break

  Determine $\sigma^{(t)}$ through solving Eq. (6.11) using the alternative weight function $W^{(alt)}(\cdot)$ defined in Eq. (6.13)

  Calculate $W(\boldsymbol{x}_i)$ for each $i = 1, ..., N$ through Eq. (6.9)

  for $K = 1, ..., K_{max}$ **do**

    Compute $\widetilde{\boldsymbol{\mu}}_K$ and BIC_K through Algorithm 8

  $\widetilde{K} = \arg\min_K BIC_K$

  $\widetilde{\boldsymbol{\mu}}^{(t)} \leftarrow \widetilde{\boldsymbol{\mu}}_{\widetilde{K}}$

  $t \leftarrow t + 1$

$T \leftarrow t - 1$

Use $h(\boldsymbol{x}; \widehat{\boldsymbol{v}^{(T)}})$ as the IS distribution and calculate the IS estimator $\widehat{p}_f$

Output: $\widehat{p}_f$

6.4.4 Component importance measures from the BiCE-CM algorithm

In the field of network reliability assessment, component importance (CI) measures are employed for ranking components based on their influence on the system failure probability. Commonly used CI measures for binary systems include among others Birnbaum's measure, critical importance factor, risk achievement worth, and Fussel-Vesely measure [50]. These measures can be extended to multi-state or continuous systems [49], e.g., after introducing a performance function $g_i(\cdot)$ at the component level [63], i.e., the i-th component fails when $g_i(x_i) \leqslant 0$.

The samples from the BiCE-CM method can be used for calculating these CI measures. Taking Birnbaum's measure (BM) as an example, it is defined as the partial derivative of the system failure probability $p_f \triangleq \Pr(g(\boldsymbol{X}) \leqslant 0)$ with respect to the component failure probability $p_{fi} \triangleq \Pr(g_i(X_i) \leqslant$

0):

$$
\begin{aligned}
BM_i \triangleq \frac{\partial p_f}{\partial p_{fi}} &= \Pr(g(\boldsymbol{X}) \leqslant 0 | g_i(X_i) \leqslant 0) - \Pr(g(\boldsymbol{X}) \leqslant 0 | g_i(X_i) > 0) \\
&= \frac{\Pr(g(\boldsymbol{X}) \leqslant 0, g_i(X_i) \leqslant 0)}{\Pr(g_i(X_i) \leqslant 0)} - \frac{\Pr(g(\boldsymbol{X}) \leqslant 0, g_i(X_i) > 0)}{\Pr(g_i(X_i) > 0)} \\
&= \frac{\mathbb{E}_{p_{\boldsymbol{X}}}\left[\mathbb{I}\{g(\boldsymbol{X}) \leqslant 0\}\mathbb{I}\{g_i(X_i) \leqslant 0\}\right]}{p_{fi}} - \frac{\mathbb{E}_{p_{\boldsymbol{X}}}\left[\mathbb{I}\{g(\boldsymbol{X}) \leqslant 0\}\mathbb{I}\{g_i(X_i) > 0\}\right]}{1 - p_{fi}}.
\end{aligned} \tag{6.42}
$$

The expectation in Eq. (6.42) can be estimated through IS using the samples from the final level of the BiCE-CM method, and p_{fi} can be estimated by crude MCS with $g_i(X_i)$, which is usually cheap to evaluate. According to the definition, the larger the BM_i, the more sensitive the failure probability p_f is to the i-th component, and hence the higher priority the component will have when allocating the system redundancy.

6.5 Numerical examples

6.5.1 Illustration: a toy connectivity problem

We consider a small network consisting of five components. Its configuration is shown in Fig. 6.1. Each component can either fail or not fail and hence is modeled by a Bernoulli distributed random variable. The topologically most important component, component 3, is assigned a failure probability of 10^{-3}, while for all other components, the failure probability is set to $3 \cdot 10^{-2}$. The connectivity between points A and B is of interest, and we have three major modes in the failure domain: $(0, 0, 1, 1, 1)$, $(1, 1, 0, 1, 1)$, and $(1, 1, 1, 0, 0)$, corresponding to three minimal cut sets: $(1, 2)$, (3), and $(4, 5)$, respectively. The probability of each mode equals $8.46 \cdot 10^{-4}, 8.85 \cdot 10^{-4}, 8.46 \cdot 10^{-4}$, respectively, and the total failure probability equals $2.80 \cdot 10^{-3}$.

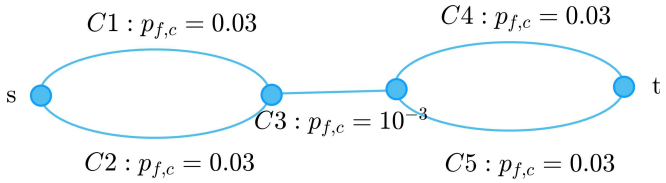
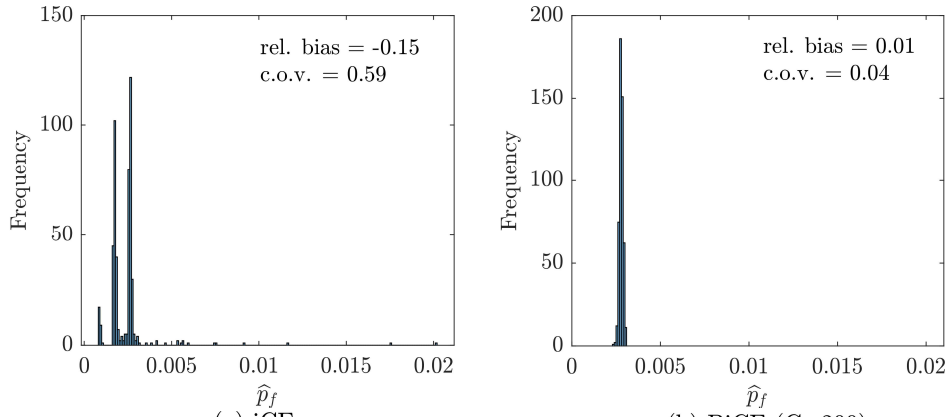


Figure 6.1: Topology of a five-component network in Example 5.1.

6.5.1.1 The zero count problem for the iCE

To illustrate the overfitting issue of the standard iCE method when solving this example, we run it 500 times with the setting $K = 3, \delta_{tar} = \delta_\epsilon = 1, N = 1000$ and plot the histogram of the 500 failure probability estimates in Fig. 6.2.

The figure illustrates a highly skewed but also multi-modal distribution of the iCE estimator. The three peaks reflect the number of cases where zero, one, or two modes are missing in the final IS



(a) iCE (b) BiCE (C=200)

Figure 6.2: Histogram of the failure probability estimates via the iCE or the BiCE-CM method. (a) results of the iCE. (b) results of the BiCE-CM.

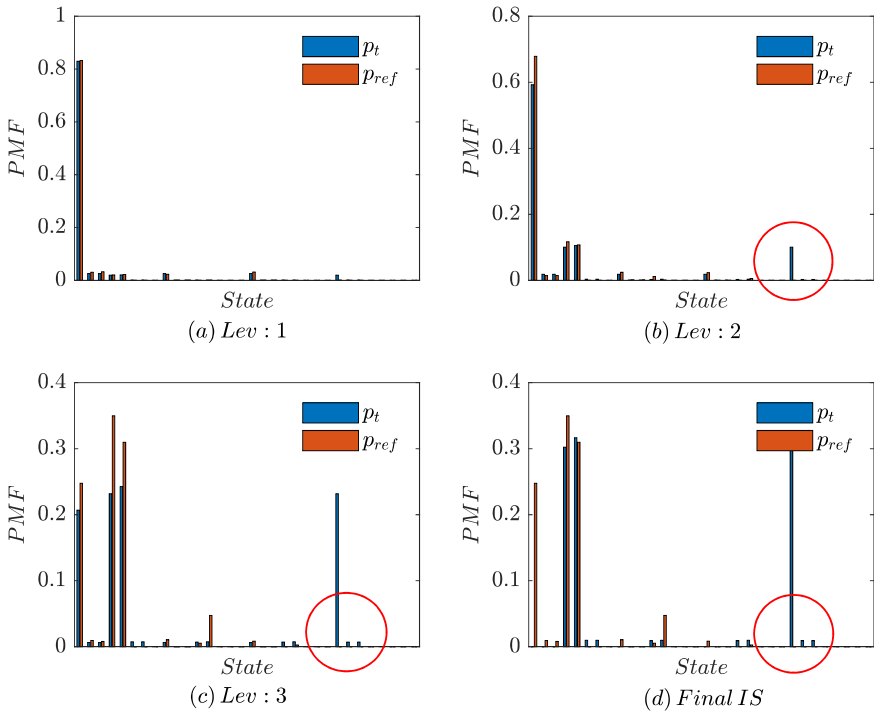
distribution. A 'missing' mode here implies that the mode is assigned a small (even zero) probability by the IS distribution. Any sample coincides with such a mode will be attached with a large weight, leading to an outlier that significantly overestimates the failure probability. By contrast, if no sample is generated from this mode, there will be a significantly negative bias. Note that the number of samples from the nominal distribution whose third component is safe follows a binomial distribution and therefore its properties can be calculated theoretically. For instance, the probability that the third component is safe for all samples generated at the first level is equal to $(1-10^{-3})^{1000} \approx 0.368$. In such case, the iCE method will definitely miss the mode (3) in all subsequent intermediate levels (see Eq. (6.22)), and the corresponding failure probability estimates will underestimate the true value, which is demonstrated in Fig. 6.2.

In Fig. 6.2(b), we show the results for the BiCE-CM method. A balanced Dirichlet prior in Eq. (6.39) is chosen for mixture parameters, with $C = 200$ and $\epsilon = 10^{-8}$. The remaining settings are the same as those of the iCE method. We can see that by introducing an appropriate prior, all three modes are found in most of the 500 estimates. A negligible relative bias (0.45%) and a small coefficient of variation (0.1) are achieved with an average of 4050 evaluations of $g(\cdot)$.

To investigate the reason for the significant difference between the performance of the two algorithms, we keep track of the reference distributions of all intermediate levels of the iCE method. The results are shown in Fig. 6.3. Fig. 6.3(a) demonstrates whether the distribution chosen at each level of the iCE method, i.e., the reference distribution, resembles the target distribution well. Apparently, the iCE method misses one of the three modes in the optimal IS distribution starting from the second level and produces a biased estimate.

6.5.1.2 Model selection or not: an empirical perspective

Next, we use the BIC for choosing adaptively the number of mixture components K at each level of the BiCE-CM. The maximum number of mixture components K_{max} is equal to 10. For comparison, we also perform the BiCE-CM method with a fixed number of K ranging from 1 to 100. Overall, 8 scenarios are considered as listed in Table 6.2. In all cases, a Dirichlet distribution is employed as a



(a) $Lev : 1$ (b) $Lev : 2$

(c) $Lev : 3$ (d) $Final\ IS$

Figure 6.3: The PMF of the target distribution and of the reference distribution at each iteration of the iCE method.

priori with $C = 200$ and $\epsilon = 10^{-8}$, and $\delta_{tar} = \delta_\epsilon$ is set to 1.

Table 6.2: Case description for example 5.1.2.

Case No.	1	2	3	4	5	6	7	8
number of mixture components, K	1	2	3	5	10	20	100	BIC

We first consider a large sample setting, where $N = 10^5$, and check the estimated KL divergence between the intractable target distribution and its mixture approximation, the reference distribution, at each level of the BiCE-CM method. The results are illustrated in Fig. 6.4. We can see from the figure that the estimated KL divergence at each intermediate level decreases as K increases, and reaches a constant minimum value at $K = 3$. This result is expected since the optimal IS distribution has three major modes and can be approximated sufficiently well by a three-component categorical mixture. Hence, additional flexibility from adding mixture components is not required. However, for $K < 3$, the model capacity is inadequate, and increasing K will lead to an IS distribution significantly closer to the optimal one thus clearly improving the performance of the BiCE-CM. Fig. 6.4 also demonstrates that selecting the K adaptively via BIC will not improve the results of a fixed K that is larger than 3, so the model selection is not needed in large sample settings for this example.

Next, we consider small sample settings, in which the weighted MLE tends to overfit the data. Although introducing a prior distribution mitigates the overfitting issue for an appropriate choice of the prior parameters, such a choice is not always straightforward. That is, a poor parameter choice

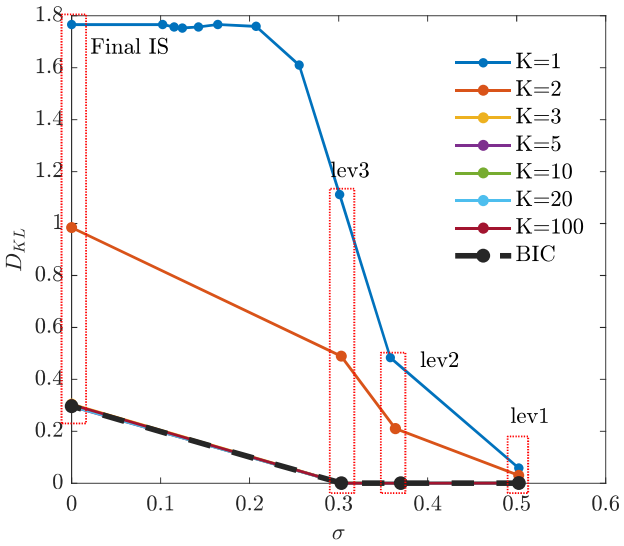


Figure 6.4: KL divergence between the intermediate target distribution and the reference distribution at each level of the BiCE-CM method (a large sample setting).

of the prior for a model with higher K could potentially result in a worse estimator. Such situations can be avoided by performing model selection. This is demonstrated by the numerical experiment, where for each scenario we run 500 times the BiCE-CM algorithm with $1,000$ samples and we set $C = 200, \epsilon = 10^{-8}$. The results are summarized through a box plot in Fig. 6.5(b).

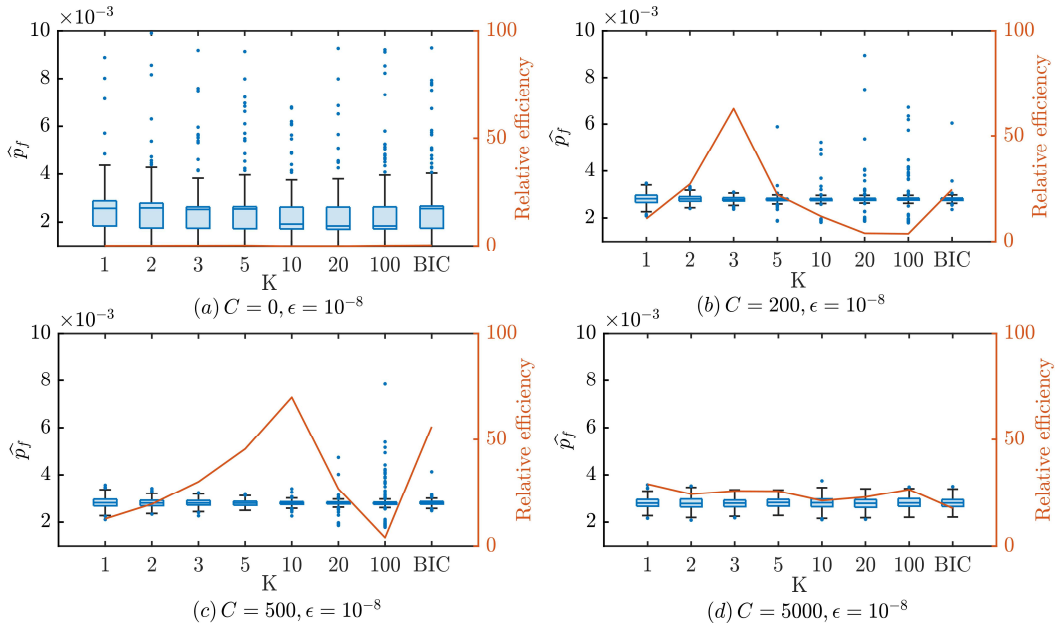


Figure 6.5: Boxplot of the estimates obtained from the BiCE-CM method. (a) $C = 0, \epsilon = 10^{-8}$, (b) $C = 200, \epsilon = 10^{-8}$, (c) $C = 500, \epsilon = 10^{-8}$, (d) $C = 5000, \epsilon = 10^{-8}$.

To measure the quality of the failure probability estimator $\widehat{p}_f$, we borrow the definition of the

'efficiency' in statistics [35], which is defined as follows

$$\mathrm{Eff}(\widehat{p}_f) \triangleq \frac{1}{\mathrm{MSE}(\widehat{p}_f) \times \mathrm{Cost}(\widehat{p}_f)}, \tag{6.43}$$

where $\mathrm{MSE}(\widehat{p}_f)$ represents the mean square error of the estimator $\widehat{p}_f$ and $\mathrm{Cost}(\widehat{p}_f)$ is the average computational cost of getting $\widehat{p}_f$, which is measured by the average number of evaluations of $g(\cdot)$ throughout all numerical examples in this paper. Note that the efficiency of the MCS equals $\frac{1}{p_f \cdot (1-p_f)}$, which is independent of the sample size. Hence, the efficiency improvement over MCS can be measured through the following relative efficiency

$$\mathrm{relEff}(\widehat{p}_f) \triangleq \frac{p_f \cdot (1-p_f)}{\mathrm{MSE}(\widehat{p}_f) \times \mathrm{Cost}(\widehat{p}_f)}. \tag{6.44}$$

The relative efficiency of different choices of K is illustrated in Fig. 6.5(b). The optimal choice, as expected, is $K = 3$. If guessing an appropriate K is not possible, adaptively selecting K via the BIC can be a good alternative. Note that this comes at a price of a significant overhead, since at each iteration, the generalized EM algorithm is performed $K_{max} = 10$ times, while for a fixed K, we only perform one single run of the algorithm. Nevertheless, for a computationally demanding performance function $g(\cdot)$, the computational cost is dominated by the evaluation of $g(\cdot)$ and the overhead resulting from the adaptive selection of K via the BIC should not be critical.

6.5.1.3 Impact of the prior distribution

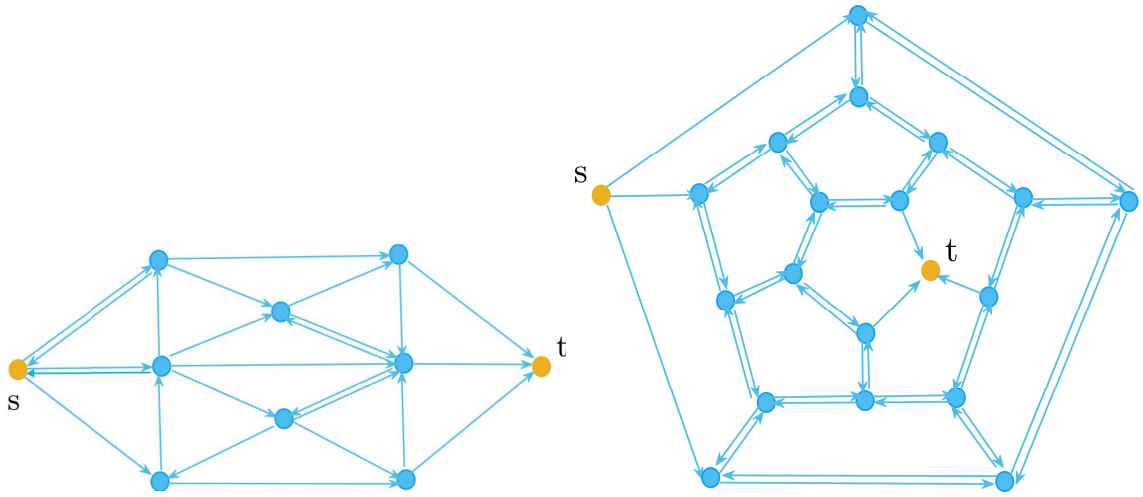
In this subsection, we study the influence of the prior distribution on the performance of the BiCE-CM method. We consider 4 different values of C, namely $0, 200, 500$ and $5,000$. ϵ is fixed at 10^{-8} for all 4 cases. The results are summarized in Fig. 6.5. When $C = 0$, the BiCE-CM method degenerates to the standard iCE method that employs the weighted MLE to update the mixture model. Due to overfitting, the relative efficiency is poor. When $C = 5000$, the weighted log-likelihood function is over-penalized, and the prior estimate dominates the data-related estimate in Eq. (6.35). Owing to the symmetric Dirichlet prior, the resulting IS distribution is close to an independent uniform distribution, and the BiCE-CM with different K performs similarly. For this 5-component toy example, an independent uniform distribution works well, however, as will be shown later, this is not generally the case. When C is appropriately large, the performance of the BiCE-CM method is shown in Fig. 6.5(b-c), and has been discussed in Subsec. 5.1.2.

6.5.2 Comparison: a benchmark study

In this subsection, we consider the multi-state two-terminal reliability problems [29], in which we compute the probability that a specified amount of 'flow' can (or cannot) be delivered from the source to the sink. This problem has been extensively studied in operations research [48, 29, 59, 6, 9], from which we borrow two benchmark problems, namely the Fishman network and the Dodecahedron network, to test the performance of the BiCE-CM method. The results are further compared with the creation-process-based splitting (CP-splitting) [9], which is a state-of-art technique for solving multi-state two-terminal reliability problems, especially when the failure probability p_f is small.

The network topology of the two benchmarks is illustrated in Fig 6.6, and we employ the same problem settings as in [9]. We consider only the edge capacities, each following an independently and identically distributed categorical distribution. Following this distribution the probability of each edge capacity being $0, 100, 200$ equals $p_0, \frac{1-p_0}{2}, \frac{1-p_0}{2}$ respectively. We are interested in the probability that the maximum flow from the source node s to the sink node t is less or equal to the threshold thr, i.e., $\Pr(\mathrm{mf}(s,t) \leqslant thr)$. We estimate this probability for each combination of $p_0 \in \{10^{-3}, 10^{-4}\}$ and $thr \in \{0, 100\}$, and for each of the two benchmarks. The reference failure probability p_{ref} in each scenario is calculated using the CP-splitting method with 10^6 trajectories. The results are summarized in Table 6.3 and 6.4.

For the BiCE-CM method, we set $N = 2000, \delta_{tar} = \delta_\epsilon = 1.5, C = 200, \epsilon = 10^{-8}$, and compute the mean value, c.o.v., the average number of evaluations of $g(\cdot)$, and the relative efficiency through 500 independent repetitions of the algorithm. For the CP-splitting method, we report the results from Tables 3 and 4 in [9]. Therein, the c.o.v. is computed for the mean value of 1000 repetitions. To obtain the c.o.v. of a single repetition, which guarantees a fair comparison between the two methods, the c.o.v. reported in [9] is multiplied by $\sqrt{1,000}$. In addition, the number of $g(\cdot)$ evaluations in CP-splitting is computed by multiplying the number of levels by the number of trajectories, without considering the pilot run.



(a) Fishman network. (b) Dodecahedron network.

Figure 6.6: Topology of the two benchmarks in Example 5.2.

The performance of the BiCE-CM method for the two benchmarks is demonstrated in Table 6.3 and 6.4, in which the results of the CP-splitting method are enclosed in the parentheses for comparison.

From these two tables, we observe a clear variance reduction in the BiCE-CM estimator without increasing the computational cost compared to the CP-splitting method. The standard iCE performs poorly for these two benchmarks due to the choice of a small p_0.

Fig. 6.7 illustrates the impact of different prior parameters C and of different K on the performance of the BiCE-CM method. We herein consider the Dodecahedron network with $thr = 0$ and $p_0 = 10^{-3}$. When $C = 5000$, the prior estimate dominates the data-related estimate in Eq. (6.33) and results

Table 6.3: Performance of the BiCE method for the Fishman network in example 5.2.

	p_{ref}	mean	c.o.v.	cost	relEff
$p_0 : 10^{-3}, thr : 100$	$3.00 \cdot 10^{-6}$	$3.03(3.00^*) \cdot 10^{-6}$	0.05(0.17)	$1.03(0.90) \cdot 10^{4}$	$1.2(0.13) \cdot 10^{4}$
$p_0 : 10^{-4}, thr : 100$	$3.00 \cdot 10^{-8}$	$3.01(3.00) \cdot 10^{-8}$	0.04(0.21)	$1.40(1.30) \cdot 10^{4}$	$1.5(0.058) \cdot 10^{6}$
$p_0 : 10^{-3}, thr : 0$	$2.03 \cdot 10^{-9}$	$2.01(2.02) \cdot 10^{-9}$	0.04(0.24)	$1.40(1.40) \cdot 10^{4}$	$2.1(0.062) \cdot 10^{7}$
$p_0 : 10^{-4}, thr : 0$	$2.00 \cdot 10^{-12}$	$2.00(2.00) \cdot 10^{-12}$	0.04(0.28)	$1.80(1.80) \cdot 10^{4}$	$1.7(0.035) \cdot 10^{10}$

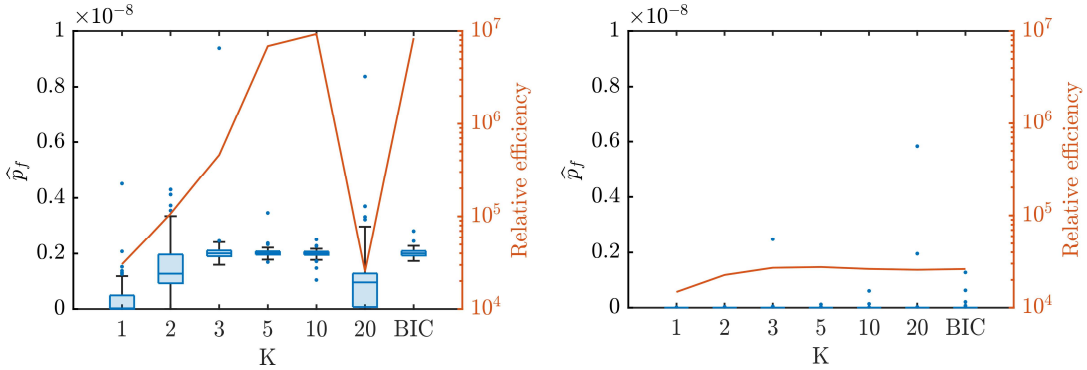
* The number in the parentheses shows the result of the CP-splitting method.

Table 6.4: Performance of the BiCE method for the Dodecahedron network in example 5.2.

	p_{ref}	mean	c.o.v.	cost	relEff
$p_0 : 10^{-3}, thr : 100$	$3.05 \cdot 10^{-6}$	$3.04(3.03^*) \cdot 10^{-6}$	0.06(0.20)	$1.11(0.90) \cdot 10^{4}$	$8.2(0.92) \cdot 10^{3}$
$p_0 : 10^{-4}, thr : 100$	$3.08 \cdot 10^{-8}$	$3.00(2.99) \cdot 10^{-8}$	0.06(0.23)	$1.40(1.30) \cdot 10^{4}$	$7.6(0.49) \cdot 10^{5}$
$p_0 : 10^{-3}, thr : 0$	$2.06 \cdot 10^{-9}$	$2.01(2.03) \cdot 10^{-9}$	0.05(0.26)	$1.41(1.30) \cdot 10^{4}$	$1.2(0.057) \cdot 10^{7}$
$p_0 : 10^{-4}, thr : 0$	$2.02 \cdot 10^{-12}$	$1.99(1.97) \cdot 10^{-12}$	0.06(0.27)	$1.80(2.10) \cdot 10^{4}$	$7.4(0.34) \cdot 10^{9}$

* The number in the parentheses shows the result of the CP-splitting method.

in a near uniform IS distribution. In such cases, the performance of the BiCE-CM is poor. On the contrary, when $C = 200$, which is a minor proportion of the N, the BiCE-CM works well for K equal to 5 or 10 or when employing BIC.



(a) $C = 200, \epsilon = 10^{-8}$ (b) $C = 5000, \epsilon = 10^{-8}$

Figure 6.7: Boxplot of the BiCE-CM estimates for the Dodecahedron network with $thr = 0$ and $p_0 = 10^{-3}$. (a) $C = 200, \epsilon = 10^{-8}$, (b) $C = 5000, \epsilon = 10^{-8}$.

6.5.3 Application: the IEEE 30 benchmark model with common cause failure

In this subsection, we consider the IEEE 30 power transmission network [60] illustrated in Fig. 6.8. The network consists of 6 power generators, 24 substations, and 41 transmission lines, which we assume to be subjected to earthquakes.

The hypocenter of the earthquake is assumed to be fixed and the earthquake magnitude is described

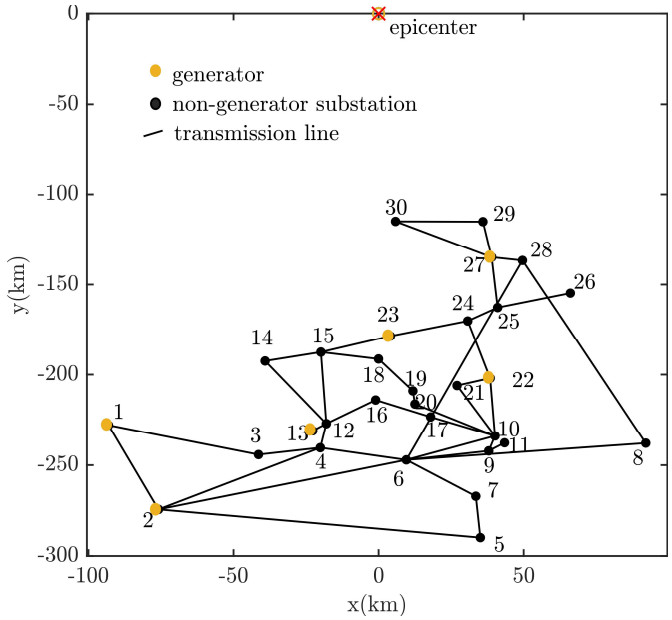


Figure 6.8: Network topology of the IEEE30 benchmark.

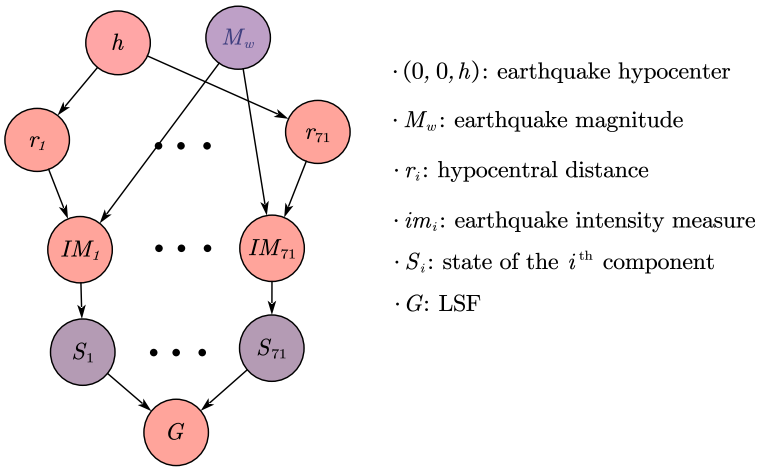


Figure 6.9: Dependence structure for the IEEE30 benchmark subjected to earthquakes. The purple nodes represent the random variables.

by a truncated exponential distribution $p_M \propto \exp(-0.85m), \quad 5 \leqslant m \leqslant 8$. The failures of the network components are dependent as they occur due to the earthquake, but it is often assumed that they are conditional independent given the earthquake [52]. Such conditional independence is depicted in Fig. 6.9 [66], where r_i represents the hypocentral distance of the i-th component, and im_i is the intensity measure of i. In the present example, im_i is a deterministic function of r_i described by the ground motion predictive equation (GMPE) given in [21]. S_i denotes the state of the component i, whose distribution is indicated by the fragility curves in [11]. For each of the 6 generators, we consider 5 damage states, namely negligible, minor, moderate, extensive, and complete damage, which correspond to 0%, 20%, 60%, 80%, and 100% reduction of power production, respectively. The remaining 24 non-generator buses and all 41 transmission branches have 2 damage states, either safe or complete failure. The distribution of different network components is summarized in Table 6.5.

Table 6.5: The distribution of different components for the IEEE30 benchmark.

	generators	non-generator buses	transmission lines
# components	6	24	41
distribution	categorical	Bernoulli	Bernoulli
reference	Table 6.6 in [11]	Table 6.9 in [11]	$p_f = 5 \cdot 10^{-2}$

We measure the network performance by the load shedding based on a direct current optimal power flow (DC-OPF) analysis using MATPOWER v7.1 [60]. The system failure is defined as over 50% of the total power demand being shed after the earthquake, which gives the following performance function:

$$g(\boldsymbol{x}) \triangleq 50\% - \frac{LS(\boldsymbol{x})}{D_{tot}}, \tag{6.45}$$

where $LS(\boldsymbol{x})$ represents the load shedding with the network configuration, or state, $\boldsymbol{x}$, and D_{tot} is the total power demand. The failure probability approximated by one single crude MCS with 10^6 samples is equal to 0.0013, which is then employed as the reference for validating the proposed BiCE-CM algorithm. For the BiCE-CM, 200 independent runs with $N = 2,000, \delta_{tar} = \delta_\epsilon = 1.5$ are launched, based on which, we calculate the mean, c.o.v. and the relative efficiency of the BiCE-CM estimator. The number of mixture components K is adaptively chosen via the BIC, and we investigate 4 different prior distributions with $C \in \{0, 200, 400, 5000\}$ and $\epsilon = 10^{-8}$. The results are depicted in Fig. 6.10, where it is shown that the BiCE-CM with $C = 400$ performs the best among the four investigated cases. In particular, it significantly outperforms the $C = 0$ case, which represents the standard iCE method. The relative efficiency of the BiCE-CM with $C = 400$ is about 6, meaning the efficiency is around 6 times higher than that of the crude MCS. The average CPU time of the BiCE-CM is as 371.23 seconds on a 3.50GHz Intel Xeon E3-1270v3 computer. As a comparison, crude MCS needs 46,161 samples to achieve the same coefficient of variation as the BiCE-CM, which takes 1741.68 seconds on the same computer. Hence, the overhead of BiCE-CM does not strongly affect the overall computation time.

The BM averaged over 200 repetitions of the BiCE-CM algorithm is depicted in Fig. 6.11 for different components of the IEEE30 benchmark model. For multi-state generators, the failure is defined as the power production being reduced by 80% or more. We can see from the figure that except for components 3,4 and 8, the BM evaluated with the BiCE-CM method is consistent with that evaluated by crude MCS.

6.6 Conclusions

In network reliability assessments, the network components are often strongly dependent given system failure. Such dependence cannot be captured by the independent categorical distribution employed in the original Bayesian improved cross entropy (BiCE) paper. To capture this dependence and improve the performance of the estimate, we employ instead the categorical mixture as the parametric family of the BiCE. The parameters of the mixture model are updated through the weighted maximum a posteriori (MAP) estimate. In this way, the overfitting issue encountered in the standard improved cross entropy (iCE) method, which employs the weighted maximum likelihood

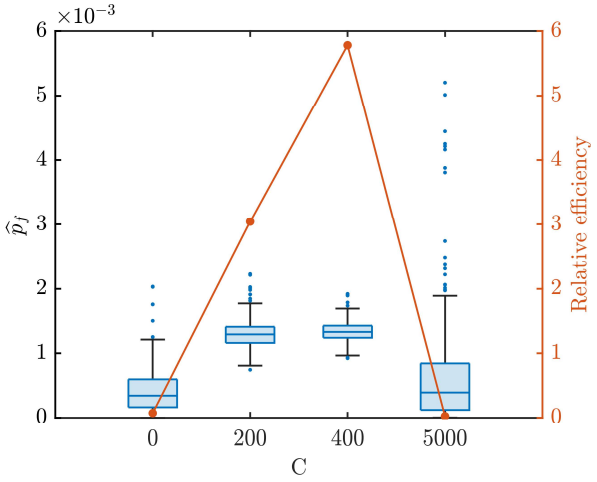


Figure 6.10: Boxplot of the BiCE-CM estimates for the IEEE30 benchmark model.

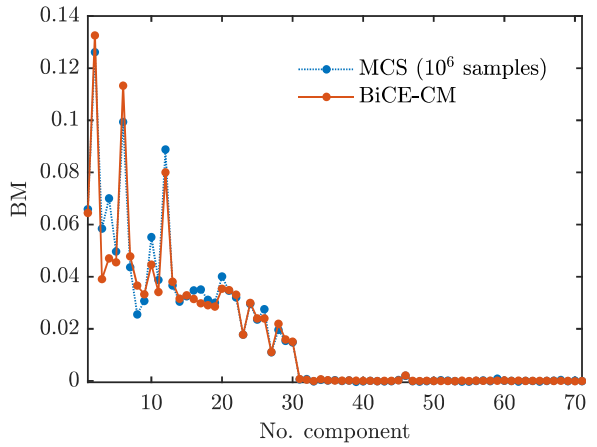


Figure 6.11: Birnbaum's measure for different components of the IEEE30 benchmark.

estimate (MLE), is mitigated. The proposed algorithm is termed the BiCE-CM method.

We approximate the weighted MAP through the expectation maximization(EM) algorithm with a minor modification to account for the weights and the prior. The algorithm results in a monotonically increasing weighted posterior and converges to a local maximum, a saddle point, or a boundary point depending on the starting point of the generalized EM algorithm. Moreover, the Bayesian information criterion (BIC) can be computed as a by-product of the generalized EM algorithm and is employed as model selection technique for choosing the optimal number of components in the mixture when the sample size is moderate. The model selection technique is unnecessary in a large sample setting in which case a large number of mixture components is suggested. A set of numerical examples demonstrates that the proposed algorithm outperforms the standard iCE and the BiCE with the independent categorical distribution. Note that there is no guarantee that the BiCE-CM can find all major failure modes. The accuracy and efficiency of the BiCE-CM depend highly on the choice of the prior distribution. In this paper, we suggest a balanced prior that works well in all our numerical examples. A detailed investigation of alternative choices of the prior should be carried out. In addition, the BiCE-CM method does not directly apply to high dimensional problems due to the degeneration of the IS weights, and hence, dimensionality reduction techniques should be

employed in such cases. These two aspects will be addressed in future work.

6.7 Acknowledgment

The first author gratefully acknowledges the financial support of the China Scholarship Council.

References

[1] C. Alexopoulos. "State space partitioning methods for stochastic shortest path problems". In: *Networks: An International Journal* 30.1 (1997), pp. 9–21.

[2] M. O. Ball, C. J. Colbourn, and J. S. Provan. "Network reliability". In: *Handbooks in Operations Research and Management Science* 7 (1995), pp. 673–762.

[3] M. O. Ball and J. S. Provan. "Disjoint products and efficient computation of reliability". In: *Operations Research* 36.5 (1988), pp. 703–715.

[4] J.-P. Baudry and G. Celeux. "EM for mixtures". In: *Statistics and Computing* 25.4 (2015), pp. 713–726.

[5] Z. I. Botev, P. L'Ecuyer, G. Rubino, R. Simard, and B. Tuffin. "Static network reliability estimation via generalized splitting". In: *INFORMS Journal on Computing* 25.1 (2013), pp. 56–71.

[6] Z. I. Botev, P. L'Ecuyer, and B. Tuffin. "Reliability estimation for networks with minimal flow demand and random link capacities". In: *arXiv preprint arXiv:1805.03326* (2018).

[7] F. Cadini, G. L. Agliardi, and E. Zio. "Estimation of rare event probabilities in power transmission networks subject to cascading failures". In: *Reliability Engineering & System Safety* 158 (2017), pp. 9–20.

[8] H. Cancela, M. El Khadiri, and G. Rubino. "A new simulation method based on the RVR principle for the rare event network reliability problem". In: *Annals of Operations Research* 196 (2012), pp. 111–136.

[9] H. Cancela, L. Murray, and G. Rubino. "Efficient estimation of stochastic flow network reliability". In: *IEEE Transactions on Reliability* 68.3 (2019), pp. 954–970.

[10] H. Cancela, L. Murray, and G. Rubino. "Reliability estimation for stochastic flow networks with dependent arcs". In: *IEEE Transactions on Reliability* (2022), pp. 622–636.

[11] F. Cavalieri, P. Franchin, and P. E. Pinto. "Fragility functions of electric power stations". In: *Typology Definition and Fragility Functions for Physical Elements at Seismic Risk*. Springer, 2014, pp. 157–185.

[12] J. Chan, I. Papaioannou, and D. Straub. "An adaptive subset simulation algorithm for system reliability analysis with discontinuous limit states". In: *Reliability Engineering & System Safety* 225 (2022), p. 108607.

[13] J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method for network reliability assessment". In: *Structural Safety* 103 (2023), p. 102344.

[14] J. Chan, I. Papaioannou, and D. Straub. "Improved cross-entropy-based importance sampling for network reliability assessment". In: *Proceedings of the 13th International Conference on Structural Safety & Reliability*. ICOSSAR. 2022.

[15] J. C. Chan and D. P. Kroese. "Improved cross-entropy method for estimation". In: *Statistics and Computing* 22.5 (2012), pp. 1031–1040.

[16] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. "A tutorial on the cross-entropy method". In: *Annals of Operations Research* 134 (2005), pp. 19–67.

[17] N. L. Dehghani, S. Zamanian, and A. Shafieezadeh. "Adaptive network reliability analysis: Methodology and applications to power grid". In: *Reliability Engineering & System Safety* 216 (2021), p. 107973.

[18] P. Doulliez and E. Jamoulle. "Transportation networks with random arc capacities". In: *Revue Française d'Automatique, Informatique, Recherche Opérationnelle. Recherche Opérationnelle* 6.V3 (1972), pp. 45–59.

[19] L. Duenas-Osorio, K. Meel, R. Paredes, and M. Vardi. "Counting-based reliability estimation for power-transmission grids". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[20] T. Elperin, I. Gertsbakh, and M. Lomonosov. "Estimation of network reliability using graph evolution models". In: *IEEE Transactions on Reliability* 40.5 (1991), pp. 572–581.

[21] L. Esteva and R. Villaverde. "Seismic risk, design spectra and structural reliability". In: *Proceedings of the 5th World Conference on Earthquake Engineering*. Vol. 2. 1973, pp. 2586–2596.

[22] G. S. Fishman. "A Monte Carlo sampling plan for estimating network reliability". In: *Operations Research* 34.4 (1986), pp. 581–594.

[23] S. Frühwirth-Schnatter. "Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques". In: *The Econometrics Journal* 7.1 (2004), pp. 143–167.

[24] S. Frühwirth-Schnatter, G. Celeux, and C. P. Robert. *Handbook of Mixture Analysis*. CRC press, 2019.

[25] A. E. Gelfand and D. K. Dey. "Bayesian model choice: Asymptotics and exact calculations". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 56.3 (1994), pp. 501–514.

[26] S. Geyer, I. Papaioannou, and D. Straub. "Cross-entropy-based importance sampling using Gaussian densities revisited". In: *Structural Safety* 76 (2019), pp. 15–27.

[27] G. Hardy, C. Lucet, and N. Limnios. "K-terminal network reliability measures with binary decision diagrams". In: *IEEE Transactions on Reliability* 56.3 (2007), pp. 506–515.

[28] K.-P. Hui, N. Bean, M. Kraetzl, and D. P. Kroese. "The cross-entropy method for network reliability estimation". In: *Annals of Operations Research* 134.1 (2005), pp. 101–118.

[29] C.-C. Jane and Y.-W. Laih. "A practical algorithm for computing multi-state two-terminal reliability". In: *IEEE Transactions on Reliability* 57.2 (2008), pp. 295–302.

[30] H. A. Jensen and D. J. Jerez. "A stochastic framework for reliability and sensitivity analysis of large scale water distribution networks". In: *Reliability Engineering & System Safety* 176 (2018), pp. 80–92.

[31] W.-H. Kang, J. Song, and P. Gardoni. "Matrix-based system reliability method and applications to bridge networks". In: *Reliability Engineering & System Safety* 93.11 (2008), pp. 1584–1593.

[32] B. Kaynar and A. Ridder. "The cross-entropy method with patching for rare-event simulation of large Markov chains". In: *European Journal of Operational Research* 207.3 (2010), pp. 1380–1397.

[33] A. Kong. "A note on importance sampling using standardized weights". In: *University of Chicago, Dept. of Statistics, Tech. Rep* 348 (1992).

[34] N. Kurtz and J. Song. "Cross-entropy-based adaptive importance sampling using Gaussian mixture". In: *Structural Safety* 42 (2013), pp. 35–44.

[35] P. L'Ecuyer. "Efficiency improvement and variance reduction". In: *Proceedings of Winter Simulation Conference*. IEEE. 1994, pp. 122–132.

[36] J. Li. "A PDEM-based perspective to engineering reliability: From structures to lifeline networks". In: *Frontiers of Structural and Civil Engineering* 14.5 (2020), pp. 1056–1065.

[37] J. Li and J. He. "A recursive decomposition algorithm for network seismic reliability evaluation". In: *Earthquake Engineering & Structural Dynamics* 31.8 (2002), pp. 1525–1539.

[38] H.-W. Lim and J. Song. "Efficient risk assessment of lifeline networks under spatially correlated ground motions using selective recursive decomposition algorithm". In: *Earthquake Engineering & Structural Dynamics* 41.13 (2012), pp. 1861–1882.

[39] J.-S. Lin, C.-C. Jane, and J. Yuan. "On reliability evaluation of a capacitated-flow network in terms of minimal pathsets". In: *Networks* 25.3 (1995), pp. 131–138.

[40] A. Lisnianski and G. Levitin. *Multi-state System Reliability: Assessment, Optimization and Applications*. World scientific, 2003.

[41] X.-L. Meng and W. H. Wong. "Simulating ratios of normalizing constants via a simple identity: A theoretical exploration". In: *Statistica Sinica* (1996), pp. 831–860.

[42] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[43] L. Murray, H. Cancela, and G. Rubino. "A splitting algorithm for network reliability estimation". In: *IIE Transactions* 45.2 (2013), pp. 177–189.

[44] I. Papaioannou, S. Geyer, and D. Straub. "Improved cross-entropy-based importance sampling with a flexible mixture model". In: *Reliability Engineering & System Safety* 191 (2019), p. 106564.

[45] R. Paredes, L. Dueñas-Osorio, and I. Hernandez-Fajardo. "Decomposition algorithms for system reliability estimation with applications to interdependent lifeline networks". In: *Earthquake Engineering & Structural Dynamics* 47.13 (2018), pp. 2581–2600.

[46] R. Paredes, L. Dueñas-Osorio, K. S. Meel, and M. Y. Vardi. "Principled network reliability approximation: A counting-based approach". In: *Reliability Engineering & System Safety* 191 (2019), p. 106472.

[47] J. S. Provan and M. O. Ball. "Computing network reliability in time polynomial in the number of cuts". In: *Operations Research* 32.3 (1984), pp. 516–526.

[48] J. E. Ramirez-Marquez and D. W. Coit. "A Monte Carlo simulation approach for approximating multi-state two-terminal reliability". In: *Reliability Engineering & System Safety* 87.2 (2005), pp. 253–264.

[49] J. E. Ramirez-Marquez and D. W. Coit. "Composite importance measures for multi-state systems with multi-state components". In: *IEEE Transactions on Reliability* 54.3 (2005), pp. 517–529.

[50] M. Rausand and A. Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications*. John Wiley & Sons, 2003.

[51] K. Roeder and L. Wasserman. "Practical Bayesian density estimation using mixtures of normals". In: *Journal of the American Statistical Association* 92.439 (1997), pp. 894–902.

[52] H. Rosero-Velásquez and D. Straub. "Selection of representative natural hazard scenarios for engineering systems". In: *Earthquake Engineering & Structural Dynamics* 51.15 (2022), pp. 3680–3700.

[53] J. Rousseau and K. Mengersen. "Asymptotic behaviour of the posterior distribution in overfitted mixture models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.5 (2011), pp. 689–710.

[54] G. Schwarz. "Estimating the dimension of a model". In: *The Annals of Statistics* (1978), pp. 461–464.

[55] J. Song and W.-H. Kang. "System reliability and sensitivity under statistical dependence by matrix-based system reliability method". In: *Structural Safety* 31.2 (2009), pp. 148–156.

[56] R. J. Steele and A. E. Raftery. "Performance of Bayesian model selection criteria for Gaussian mixture models". In: (2010).

[57] F. Uribe, I. Papaioannou, Y. M. Marzouk, and D. Straub. "Cross-entropy-based importance sampling with failure-informed dimension reduction for rare event simulation". In: *SIAM/ASA Journal on Uncertainty Quantification* 9.2 (2021), pp. 818–847.

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

[59] W.-C. Yeh. "An improved sum-of-disjoint-products technique for symbolic multi-state flow network reliability". In: *IEEE Transactions on Reliability* 64.4 (2015), pp. 1185–1193.

[60] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education". In: *IEEE Transactions on Power Systems* 26.1 (2010), pp. 12–19.

[61] E. Zio. *Monte Carlo Simulation: The Method*. Springer, 2013.

[62] E. Zio and N. Pedroni. "Reliability analysis of discrete multi-state systems by means of subset simulation". In: *Proceedings of the 17th ESREL Conference*. 2008, pp. 22–25.

[63] E. Zio and L. Podofillini. "Importance measures of multi-state components in multi-state systems". In: *International Journal of Reliability, Quality and Safety Engineering* 10.03 (2003), pp. 289–310.

[64] K. M. Zuev, S. Wu, and J. L. Beck. "General network reliability problem and its efficient solution by subset simulation". In: *Probabilistic Engineering Mechanics* 40 (2015), pp. 25–35.

[65] M. J. Zuo, Z. Tian, and H.-Z. Huang. "An efficient method for reliability evaluation of multistate networks given all minimal path vectors". In: *IIE Transactions* 39.8 (2007), pp. 811–817.

[66] K. Zwirglmaier, J. Chan, I. Papaioannou, J. Song, and D. Straub. "Hybrid Bayesian networks for reliability assessment of infrastructure systems". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* (2023).

Chapter 7

Adaptive Monte Carlo methods for estimating rare events in power grids

Original Publication

J. Chan, R. Paredes, I. Papaioannou, L. Duenas-Osorio, and D. Straub. "Adaptive Monte Carlo methods for estimating rare events in power grids". (Under review).

Abstract

This paper presents a comprehensive study on rare event estimation in power grids, focusing on state-of-the-art adaptive Monte Carlo algorithms. We compare these methods for the optimal power flow problem in various IEEE benchmark models. Based on the results of our study, we analyze the pros and cons of each adaptive method and investigate their beneficial combinations. Overall, the adaptive effort subset simulation (aE-SuS) method and particle integration methods (PIMs) are promising for high-dimensional reliability analysis. By building on IEEE benchmarks, we provide fair examples for comparing different emerging methods in static network reliability assessment while revealing improvements for these methods. In particular, we introduce a hybrid approach that combines the strengths of both aE-SuS and annealed PIM. Although this method is not as efficient as aE-SuS, it significantly outperforms crude Monte Carlo and is unbiased. We then employ the aE-SuS method and this hybrid approach for risk assessment of the Texas synthetic power grid, which comprises over 5,000 components.

7.1 Introduction

Accurate and efficient network reliability assessment forms the basis of reliability-based decision making, network optimization, and community resilience. One fundamental problem is to compute or estimate the failure probability p_f, the probability that a network performance metric exceeds a specified failure threshold γ. The network performance can be described by a function $g(\cdot)$, known as performance function. In particular, let $\boldsymbol{X}$ describe the state of network components, whose sample space is $\Omega_{\boldsymbol{X}}$, and whose probability mass function (PMF) is $p_{\boldsymbol{X}}(\boldsymbol{x})$. $\mathbb{I}\{\cdot\}$ denotes the indicator function that takes value one when the statement inside the braces is true and zero otherwise. The failure probability can then be written as:

$$\begin{aligned} p_f &\triangleq \mathbb{E}_{\boldsymbol{X}}[\mathbb{I}\{g(\boldsymbol{X}) \geqslant \gamma\}] \\ &= \sum_{\boldsymbol{x}\in\Omega_{\boldsymbol{X}}} \mathbb{I}\{g(\boldsymbol{x}) \geqslant \gamma\} p_{\boldsymbol{X}}(\boldsymbol{x}), \end{aligned} \tag{7.1}$$

with failure defined as $F \triangleq \{g(\boldsymbol{X}) \geqslant \gamma\}$. When quantifying the reliability of a power system, $\boldsymbol{X}$ can represent the damage state of transmission lines and/or connecting buses, and the system performance can be either connectivity or power flow. In such settings, $\boldsymbol{X}$ usually contains discrete random variables, which can result in a discontinuous distribution of the network performance.

A set of efficient non-sampling methodologies are applicable for solving such kinds of problems [30, 33, 23]. Many of them rely on specific assumptions, such as binary states, independent components, perfect nodes, and coherent systems, which limit their generality. Other popular non-sampling-based methods, such as the matrix-based system reliability assessment [9], are restricted to a small or moderate number of components. Sampling-based methods, including crude Monte Carlo simulation (MCS) [38, 43] and its different variants, trade efficiency and accuracy for their broad applicability. These methods are often non-intrusive and treat the network model as a black box, which facilitates the use of advanced network models in the analysis. Nevertheless, the accuracy of these methods depends on the number of samples and the problems at hand, which highlights the importance of testing different emerging algorithms through unified benchmarks. For rare event estimation, crude MCS is infeasible when the limit state function is expensive to compute, and hence, advanced variance reduction techniques or meta-models have been proposed. These techniques include the standard subset simulation (SuS) [44, 46, 27], various creation process embedded methods [25, 7, 11], and actively trained meta models [10, 19].

Recently, Chan et al. [14, 15, 17] generalized widely used structural reliability algorithms and proposed the adaptive effort subset simulation method (aE-SuS) and Bayesian improved cross entropy (BiCE) method for sampling efficiently in discrete space. Concurrently, Paredes et al. [34, 35] built upon particle integration methods (PIMs) [21] and the Gamma Bernoulli approximation scheme (GBAS) [24] for probably approximately correct estimates.

The present paper tests the performance of these algorithms for rare event estimation in power grids. In particular, we consider applications on the optimal direct current (DC) power flow of IEEE benchmark models, which are extensively used in the electricity-market applications, network expansion planning, and contingency analyses [29, 26, 39]. The benchmark study provides unified examples for comparing different emerging methods in static network reliability assessment while unraveling the improvement or combination of these methods for larger and more challenging problems. To the

best of our knowledge, although crucial, a comprehensive comparison across the most competitive methods is still absent.

Building on the results of the comparative study, we further introduce a hybrid approach that combines the strengths of both aE-SuS and PIMs, which gives an unbiased estimator even in problems where significant discontinuity is involved in the network performance. Although it is less efficient (See Eq. (7.8) for the definition of efficiency) than aE-SuS, if taking into account the pilot cost, it significantly outperforms crude MCS and is unbiased. This hybrid approach and the aE-SuS method are subsequently used for risk assessment of the Texas synthetic power grid to push the boundaries of solvable problems today. This power grid represents a large-scale network consisting of over 5,000 components. Since selecting the threshold γ in Eq. (7.1) can pose a challenge, we provide the complete cumulative distribution function (CDF) of the network performance and further compute the component importance measure for each component and various thresholds. The CDF and the component importance measure can be computed using samples from one single run of the aE-SuS algorithm. Additionally, the relative efficiency [15] of the algorithm, which compares the algorithm's efficiency to that of crude MCS, is calculated as a function of the threshold γ.

The rest of the paper is organized as follows: A brief introduction of the methodologies employed in this paper is given in Section 2. In Section 3, we introduce the optimal power flow problem and conduct a comparative study. Additionally, we combine aE-SuS and PIMs to give a more robust estimator than the original PIMs. Section 4 provides a detailed illustration of applying aE-SuS for risk assessment in a large-scale synthetic power grid, along with a novel sensitivity analysis. For comparison, the results of the hybrid approach are also briefly presented. The codes for the benchmarks, as well as the adaptive Monte Carlo methods, can be found in the following repository: https://github.com/chanovo/adaptMCS-benchmarks.

7.2 Adaptive Monte Carlo methods

The adaptive Monte Carlo methods operate in an adaptive manner and rely on generating samples sequentially from a series of target distributions that gradually approach the system failure domain. This can be done either through Markov Chain Monte Carlo (MCMC) or by fitting a parametric model. The former strategy is employed in the multi-level splitting method [13], while the latter one is found in various adaptive importance sampling approaches. SuS, aE-SuS, and PIMs are special cases of multi-level splitting, and iCE is an instance of adaptive importance sampling. Consequently, these methods are referred to as the adaptive Monte Carlo methods in this study. In the following, we give a brief introduction to the adaptive Monte Carlo methods whose performance is investigated in the benchmark study.

7.2.1 Particle Integration Methods

Particle Integration Methods (PIMs) consist of sequential systems of samples, or particles, for approximating intractable integrals over Markov chains [20]. By introducing a set of T nested intermediate failure domains, denoted as $\Omega_{\boldsymbol{X}} = F_0 \supset F_1 \supset \cdots \supset F_T = F$, and employing a set of specific transition kernels $\{\mathbb{M}_t(\cdot,\cdot)\}_{t=0}^{T}$, PIMs can also be used to address the static problem of estimating

the system failure probability in Eq. (7.1) [12]. In particular, the nested domains are defined as $F_t \triangleq \{\boldsymbol{x}|L(\boldsymbol{x}) \geq l_t\}, t = 1, \cdots, T$, where $-\infty = l_0 < l_1 < \cdots < l_{T-1} < l_T = 1$ denote intermediate levels (or importance scores), and $L(\cdot)$ is a user-specified importance function that closely relates to the network performance function $g(\cdot)$. An obvious importance function is $L(\cdot) = \frac{g(\cdot)}{\gamma}$ if $\gamma > 0$. The transition kernel $\mathbb{M}_t(\cdot, \cdot)$ is also user-specified but needs to be invariant with respect to $\frac{p_{\boldsymbol{X}}(\cdot)\mathbb{I}\{\boldsymbol{x} \in F_t\}}{Z_t}$. $Z_t = \int_{\Omega_{\boldsymbol{X}}} p_{\boldsymbol{X}}(\cdot)\mathbb{I}\{\boldsymbol{x} \in F_t\}\mathrm{d}\boldsymbol{x}$ is the normalizing constant, also known as the partition function in [20], with $Z_0 = 1$ and $Z_T = p_f$. Subsequently, the failure probability can be represented as a telescoping product $p_f = \prod_{t=1}^{T} \frac{Z_t}{Z_{t-1}} = \prod_{t=1}^{T} p_t$, where $p_t \triangleq \frac{Z_t}{Z_{t-1}}, k = 1, \cdots, T$ denotes the ratio of two successive partition functions and can be estimated using the sequence of samples (or particles) in PIMs. Note that owing to the nestedness of the intermediate domains, p_t is actually the conditional probability of $\boldsymbol{X} \in F_t$, given $\boldsymbol{X} \in F_{t-1}$, denoted as $\Pr(F_t|F_{t-1})$.

The PIMs can be divided into the interacting PIM (iPIM) and the annealed PIM (aPIM) [35]. The iPIM adaptively constructs the intermediate levels (or importance scores), such that the conditional probabilities $p_1, \cdots, p_{T-1}$ are all equal to a constant value p_0. We note that the iPIM algorithm is equivalent to the SuS algorithm as presented in [1]. However, the resulting estimator is biased, and the bias can be notable, especially when the network performance presents discontinuities. In contrast, the aPIM is guaranteed to be an unbiased estimator of the true failure probability; however, it assumes the sequence of intermediate levels to be known. As suggested by Botev and Kroese [6], one can combine the approaches to a two-step meta-algorithm where the iPIM is run first to learn the sequence of levels, and then the aPIM is run second using the identified sequence, to obtain an unbiased estimator of the true failure probability. We adopt the iPIM in Algorithm 2 of Paredes et al. [35], which is a biased estimator similar to SuS. After this, we use aPIM, which is an unbiased estimator that takes the following form:

$$\hat{p}_f^{(\mathrm{aPIM})} = s^{T-1}|\mathcal{X}_T|, \tag{7.2}$$

where $|\mathcal{X}_T|$ denotes the number of particles in the last level and $s = \frac{1}{p_0}$ is known as the splitting factor; see Algorithm 1 of [35] for implementation details. This estimator is denoted as the iPIM+aPIM in the remaining part of the manuscript. Paredes et al. [35] show that, under certain conditions, optimal tuning of such a meta-algorithm is achieved by setting $p_0 \approx 0.2032$ and that popular MCMC samplers, such as the preconditioned Crank-Nicolson and modified Metropolis-Hastings algorithms, can scale well in high dimensional problems with tens of thousands of random variables.

7.2.2 Adaptive effort subset simulation method

Similar to PIMs, the standard SuS also hinges on a sequence of nested intermediate domains $\Omega_{\boldsymbol{X}} = F_0 \supset F_1 \supset \cdots \supset F_T = F$. These domains are defined as $F_t \triangleq \{g(\boldsymbol{x}) \geq l_t\}, t = 1, \cdots, T$ with $-\infty = l_0 < l_1 < \cdots < l_T = \gamma$ denoting the intermediate levels and are equivalent to those defined in Section 2.1. The failure probability p_f is then expressed as the product of the conditional probabilities, i.e., $p_f = \prod_{t=1}^{T} p_t$, where $p_t \triangleq \Pr(F_t|F_{t-1})$. The standard SuS selects the intermediate levels adaptively such that the conditional probabilities, $p_t, t = 1, \cdots T-1$, all equal a predefined constant p_0. This requires sampling from the input distribution conditional on $F_{t-1}, t = 1, ..., T$, which is accomplished by an MCMC algorithm.

Au and Wang [2] identified issues associated with employing the standard SuS for network perfor-

mance with a discrete cumulative distribution function (CDF), where a fixed intermediate failure probability p_0 can lead to an ambiguous definition of the intermediate failure domains and hence inaccurate results. Chan et al. [14] addressed this problem in the context of network reliability assessment and proposed the adaptive effort SuS (aE-SuS) algorithm that tackled this issue through adaptively choosing the intermediate failure probability p_0 and the number of samples per level N. The aE-SuS method can be combined with any MCMC algorithm. Possible choices include the adaptive conditional sampling algorithm [31] (essentially an adaptive variant of the preconditioned Crank-Nicolson algorithm used in [35]), ideally suited for high dimensional inputs, an independent Metropolis-Hasting algorithm that efficiently samples in low-dimensional discrete spaces [14], or a Gibbs sampler for performing the reliability analysis conditional on data [47]. The algorithm starts with an initial choice of p_0 and N so that at least $N \cdot p_0 \cdot tol$ seeds (or failure samples) are obtained at each level, where tol is a prescribed hyperparameter that tunes the minimal number of seeds. This leads to an adaptive estimate of the intermediate failure probabilities in terms of the failure samples and the total number of samples at the respective level. Note that the algorithm's performance heavily depends on the choice of the MCMC algorithm. If an inappropriate MCMC is selected, the final estimator is highly skewed, and estimating the mean and the variance of a highly skewed distribution is challenging since they are sensitive to rare outliers.

7.2.3 Connections among SuS, aE-SuS, and PIMs

Although developed independently in a different context, PIMs are closely related to the SuS and aE-SuS. Specifically, the standard SuS is equivalent to the iPIM, where the intermediate levels are chosen adaptively [35]. Both methods are asymptotically unbiased. The aE-SuS method is a generalized version of SuS, and hence, also of the iPIM, particularly in the case where the network performance shows significant discontinuity [14]. By contrast, the aPIM fixes the levels in advance and provides an unbiased estimator. These levels are determined through a pilot run of the iPIM in most applications. Since aE-SuS performs at least as well as SuS (or iPIM), employing aE-SuS is expected to yield superior results during pilot runs, particularly for discontinuous network performance. This motivates the idea of combining aE-SuS and aPIM to give a more robust estimator than the original PIMs, which will be detailed in Subsection 3.5. We refer to the hybrid estimator as aE-SuS+aPIM. The features of different multi-level splitting methods are summarized in Table 7.1.

Table 7.1: Different multi-level splitting methods.

	levels	conditional probabilities	unbiased	discontinuity issue
SuS(or iPIM)	adaptive	fixed, equal to p_0	no	yes
aE-SuS	adaptive	adaptive, not equal	no	no
iPIM + aPIM	fixed	fixed, equal to p_0	yes	yes
aE-SuS + aPIM*	fixed	fixed, not equal	yes	no

* A hybrid method proposed in this manuscript

7.2.4 Bayesian improved cross entropy method with categorical mixtures

The cross entropy method is an adaptive importance sampling (IS) method for rare event estimation. Different from SuS and PiMs, the method does not rely on MCMC sampling; instead, it determines the IS distribution chosen adaptively through successively approximating a sequence of intermediate target distributions, denoted as $p_{\boldsymbol{X}}^{(t)}(\cdot), t = 1, \cdots, T$, that gradually approaches the optimal IS distribution $p^*(\cdot) \propto p_{\boldsymbol{X}}(\cdot)\mathbb{I}\{\gamma - g(\cdot) \leq 0\}$. T denotes the final level of the adaptive sequence.

There are different ways of designing intermediate target distributions. Our approach is provided by the iCE method [32], an improved version of the standard cross entropy method [37]. iCE defines the sequence by smoothing the indicator function $\mathbb{I}\{\cdot\}$ in $p^*(\cdot)$ via the standard normal CDF $\Phi(\cdot)$, i.e.,

$$p_{\boldsymbol{X}}^{(t)}(\cdot) \propto p_{\boldsymbol{X}}(\cdot)\Phi\left(-\frac{\gamma - g(\cdot)}{\sigma^{(t)}}\right),\ t = 1, ..., T \tag{7.3}$$

where $\sigma^{(t)}$ is the scaling parameter. Eq. (7.3) is only known pointwise to an unknown constant, and it is challenging to sample independently from $p_{\boldsymbol{X}}^{(t)}$. To address this issue, the iCE method specifies a parametric family $h(\cdot; \boldsymbol{v})$ and iteratively determines the distribution in $h(\cdot; \boldsymbol{v})$ by minimizing an estimate of its Kullback–Leibler divergence from $p_{\boldsymbol{X}}^{(t)}(\cdot)$. This leads to successive optimization problems:

$$\widehat{\boldsymbol{v}}^{(t)} = \arg\max_{\boldsymbol{v}} \sum_{k=1}^{N} W_k^{(t)} \ln(h(\boldsymbol{x}_k; \boldsymbol{v}) \tag{7.4}$$

$$W_k^{(t)} \triangleq \frac{p_{\boldsymbol{X}}(\boldsymbol{x}_k)\Phi\left(-\frac{\gamma - g(\boldsymbol{x}_k)}{\sigma^{(t)}}\right)}{h(\boldsymbol{x}_k; \widehat{\boldsymbol{v}}^{(t-1)})},\quad \boldsymbol{x}_k \sim h(\cdot; \widehat{\boldsymbol{v}}^{(t-1)})$$

with $h(\boldsymbol{x}; \widehat{\boldsymbol{v}}^{(0)}) = p_{\boldsymbol{X}}(\boldsymbol{x})$. The parameter $\sigma^{(t)}$ is chosen adaptively such that the effective sample size of the weighted data $\{\boldsymbol{x}_k, W_k^{(t)}\}_{k=1}^N$ at each level is approximately equal. One can prove that $\sigma^{(t)}$ decreases monotonically when the input distribution $p_{\boldsymbol{X}}(\cdot)$ is discrete and the intermediate target distributions $p_{\boldsymbol{X}}^{(t)}(\cdot), t = 1, ..., T$ can be perfectly restored from the parametric family, i.e., $h(\cdot; \widehat{\boldsymbol{v}}^{(t)}) = p_{\boldsymbol{X}}^{(t)}(\cdot), t = 1, ..., T$ [17].

Eq. (7.4) indicates that $\widehat{\boldsymbol{v}}^{(t)}$ is the weighted maximum likelihood estimation of $\boldsymbol{v}$ given the data set $\{\boldsymbol{x}_k, W_k^{(t)}\}_{k=1}^N$. Hence, $\widehat{\boldsymbol{v}}^{(t)}$ may suffer from overfitting when the sample size N is small. Chan et al. [15] circumvent this issue by introducing Bayesian statistics in iCE and propose the Bayesian iCE (BiCE) method. Specifically for network reliability assessment, they introduce a symmetric Dirichlet prior for the independent categorical distribution and substitute the weighted maximum likelihood estimation $\widehat{\boldsymbol{v}}^{(t)}$ with the weighted posterior predictive estimate, or weighted maximum a posteriori estimate. To further consider the dependence among network components, Chan et al. [16] employ a more flexible categorical mixture, where the weighted maximum a posteriori can be efficiently approximated through a generalized expectation-maximization algorithm. The BiCE method for network reliability assessment is proven to be unbiased.

In addition to the parameters required by the standard iCE, i.e., the number of samples N and the parameter δ_{tar} that adjusts the convergence of $p_{\boldsymbol{X}}^{(t)}(\cdot)$, the BiCE method introduces an additional parameter b that accounts for the 'strength' of the prior. In addition, the number of clusters in the mixture, K, should also be chosen in advance if the categorical mixture model is employed.

7.3 A comparative study

In this section, we conduct a comparative analysis of the adaptive Monte Carlo methods discussed in Section 7.2. A special focus is placed on evaluating their efficiency in solving optimal power flow problems across various IEEE benchmark models. The dimensions of these benchmarks span from dozens to several hundred variables. Based on the results of the comparative study, we delve into the strengths and weaknesses of each method and explore strategies for determining intermediate levels in aPIM. While an initial comparison of the methods was presented in our conference paper [18], this manuscript addresses an even more challenging scenario with a failure probability of approximately 10^{-5}. Moreover, we propose a hybrid method that performs better than the original PIMs, especially when there are substantial discontinuities in network performance.

7.3.1 Optimal direct current power flow problem

The power flow in power transmission networks is driven by Kirchhoff's law and various operational strategies. While alternating current (AC) models can more accurately represent power flow, particularly in stability analysis, their solution is computationally challenging due to the need for iterative solutions, and they require many inputs that are not generally available when performing the reliability analysis. For this reason, we use direct current (DC) power flow models, which approximate the flow by solving a linear equation set, in which the net reactive power injection Q_i and voltage magnitude V_i at each bus are neglected. The results are less accurate than AC's for transient analyses but adequate for system reliability assessment [3]. In addition, instead of modeling cascading failures, we focus on standard DC optimal power flow (DC-OPF) problems, where we compute the optimal power operation strategy that avoids network component failure and, at the same time, minimizes a specified cost function $\mathcal{C}(\cdot)$.

Specifically, let $\boldsymbol{\eta} \triangleq \{P_i^+, P_i^-, \theta_i\}_{i=1}^{n_b}$ collect the power injection (P_i^+), power consumption (load P_i^-), and voltage angle (θ_i) at each of the n_b buses, to formulate the standard DC-OPF as follows:

$$\begin{aligned} &\min_{\boldsymbol{\eta}} \mathcal{C}(\boldsymbol{\eta}) \\ &\text{s.t.} \mathcal{G}(\boldsymbol{\eta}) = \mathbf{0} \\ &\quad \mathcal{H}(\boldsymbol{\eta}) \leqslant \mathbf{0} \\ &\quad \boldsymbol{\eta}^{(min)} \leqslant \boldsymbol{\eta} \leqslant \boldsymbol{\eta}^{(max)} \end{aligned} \tag{7.5}$$

where equality constraints in $\mathcal{G}(\boldsymbol{\eta}) = \mathbf{0}$ represent the active power balance equations, and inequality constraints in $\mathcal{H}(\boldsymbol{\eta}) \leqslant \mathbf{0}$ result from the branch flow limits, i.e., the power flow over any branch is always below its capacity. If capacity data is unavailable, we assume that the flow limit of a branch is twice the DC power flow over this branch in the intact network. The limits $\boldsymbol{\eta}^{(min)} \leqslant \boldsymbol{\eta} \leqslant \boldsymbol{\eta}^{(max)}$ include an equality constraint on the voltage angle of the reference bus $\theta^{(ref)}$ and lower and upper bound for other variables in $\boldsymbol{\eta}$. If a linear cost function is chosen, the optimization problem is linear and hence can be efficiently solved by various linear programming solvers. To this end, a positive constant cost c is associated with each unit of the power loss, and the cost function equals the

constant c multiplied by the total power loss $l_p(\boldsymbol{\eta})$, i.e.,

$$\mathcal{C}(\boldsymbol{\eta}) \triangleq c \cdot l_p(\boldsymbol{\eta}) = c \cdot \left(P^{(dem)} - \sum_{i \in \Gamma_g} P_i^+ \right), \tag{7.6}$$

where the constant $P^{(dem)}$ represents the total power demand in the intact network, and Γ_g collects the indices of all generators.

We simulate the DC-OPF across various IEEE benchmarks in this comparative study using MATPOWER v7.1 [42]. It is worth noting that the built-in solver in MATLAB sometimes encounters difficulties in locating feasible solutions, particularly within the IEEE300 benchmark. To address this issue, we relax the minimal power generation requirement when switching on the generator, allowing P_i^+ to change continuously from 0 to its maximum value. Moreover, we ignore any load that cannot be dispatched by MATPOWER v7.1. If the built-in 'dual-simplex' algorithm crashes, we adopt the 'interior-point' algorithm to solve the linear optimization problem. The resulting optimal (or minimal) total power loss $l_p^{(\min)}$ is rounded to 8 decimal places.

7.3.2 Benchmark settings

The network performance $g(\boldsymbol{x})$ is measured by the percentage blackout size $PBS(\boldsymbol{x})$, that is, the percentage of load shed in DC-OPF. $\boldsymbol{x}$ is an instance of the input random vector $\boldsymbol{X}$. We are concerned about the probability that the percentage blackout size exceeds a specified threshold γ, i.e., $p_f \triangleq \Pr(PBS(\boldsymbol{X}) \geqslant \gamma)$. The threshold is chosen such that the failure probability, p_f, is approximately equal for each benchmark; we consider two scenarios: one with $p_f \approx 10^{-4}$ and the other with $p_f \approx 10^{-5}$.

Solving the optimization problem in Eq. (7.5) with cost function in Eq. (7.6), one evaluates the minimum power loss $l_p^{(min)}(\boldsymbol{x})$ associated with the state of network components $\boldsymbol{x}$. The percentage blackout size $PBS(\boldsymbol{x})$ can then be calculated as:

$$PBS(\boldsymbol{x}) \triangleq \frac{l_p^{(\min)}(\boldsymbol{x})}{P^{(dem)}} \cdot 100. \tag{7.7}$$

For each generator, we consider four damage states, namely negligible, minor, major, and complete damage, corresponding to $0\%, 20\%, 60\%$, and 100% reduction of the power production, respectively. The remaining connecting buses and all transmission lines have two damage states, either functioning (negligible damage) or failed (complete damage).

The reference failure probability is obtained through crude MCS with 10^8 samples for the IEEE14, 30, and 57 benchmark models and with 10^7 samples for larger benchmarks. The coefficient of variation (c.o.v.) of the MCS estimators, denoted as $\delta^{(MCS)}$, along with detailed problem settings for each benchmark, are presented in Table 7.2. In addition, Table 7.3 provides a summary of the state distribution of different network components. Each generator is modeled as a categorically distributed variable, and each transmission line or connecting bus is Bernoulli distributed.

We compare the different methods by their relative efficiency with respect to crude MCS, a concept

Table 7.2: IEEE benchmark models and their reliability in two different scenarios.

	# nodes	# lines	Scenario 1			Scenario 2		
			γ	$p_f(\times 10^{-4})$	$\delta^{(MCS)}$	γ	$p_f(\times 10^{-5})$	$\delta^{(MCS)}$
IEEE 14	14	20	54.8	1.1	1%	62.9	1.0	3%
IEEE 30	30	41	40.2	1.0	1%	53.8	0.90	3%
IEEE 57	57	80	54.1	1.0	1%	63.3	1.1	3%
IEEE 118	118	186	13.8	1.0	3%	16.6	1.0	10%
IEEE 300	300	411	26.1	1.0	3%	29.1	1.0	10%

Table 7.3: Probability distribution of network components

prob. \ state / type	complete	major	minor	negligible
generator	0.01	0.19	0.3	0.5
connecting bus	0.01	/	/	0.99
trans. line	0.01	/	/	0.99

borrowed from statistics. The relative efficiency of an estimator $\widehat{p}_f$ is defined as [15]:

$$\text{relEff}(\widehat{p}_f) \triangleq \frac{p_f \cdot (1-p_f)}{\text{MSE}(\widehat{p}_f) \times \text{Cost}(\widehat{p}_f)}, \tag{7.8}$$

For adaptive Monte Carlo methods, the computational overhead is typically negligible, i.e., the CPU time of the algorithm is dominated by running the network model. Hence, the computational cost can be reasonably measured by the number of evaluations of the network performance. With the definition in Eq. (7.8), the relative efficiency of crude MCS is equal to one; the larger the relative efficiency of an estimator, the more efficient it is relative to MCS. Intuitively, this means that if the relative efficiency equals 2, the MSE of the algorithm will be half of that of crude MCS with the same cost.

In this benchmark study, we consider the following five adaptive MCS algorithms: BiCE with single categorical distribution, BiCE with the categorical mixture, aE-SuS, the iPIM, and iPIM+aPIM. For BiCE, we use $N = 2,000, \delta_{tar} = 1.5, b = 10$ and K equals either 1 or 10. For aE-SuS, we set $N = 2,000, p_0 = 0.1, tol = 0.8$, and for the iPIM and iPIM+aPIM, we select $N = 2,000$, $p_0 = 0.1$, and use the score function $\frac{PBS(\mathbf{x})}{\gamma}$. Additionally, for all multi-level splitting methods, we employ the adaptive version of the preconditioned Crank-Nicolson sampler, also known as the adaptive conditional sampler in [31].

7.3.3 Results

We report the relative efficiency computed from 200 independent runs of each method in Tables 7.4 and 7.5, each for a different scenario. It has been observed that in the majority of settings explored in this study, 200 repetitions are sufficient to obtain an accurate estimate of the empirical mean squared error (MSE). The relative bias, c.o.v, and average computation cost of each method are calculated and reported in Appendix A. Violin plots of the failure probabilities calculated by

the different algorithms are depicted in Fig. 7.1 and 7.2, where the orange solid line represents the MCS reference, and the orange cross represents the average failure probability estimate over 200 repetitions of the tested algorithms.

For aE-SuS, besides the failure probability estimate, we can also output the empirical CDF of the survival rate, defined as $100 - PBS(\boldsymbol{X})$. In other words, we estimate $\Pr(100 - PBS(\boldsymbol{X}) \leq 100 - \gamma) = \Pr(PBS(\boldsymbol{X}) \geq \gamma)$ with changing γ, and this can be done through a single run of the aE-SuS algorithm [47]. The mean, 10, and 90 percentile of the empirical CDF, obtained with 200 independent runs of the algorithm, are shown in Fig. 7.3. The mean empirical CDF estimated from aE-SuS coincides well with the MCS reference.

Table 7.4: Relative efficiency of different sampling-based methods (Scenario 1, $p_f \approx 10^{-4}$).

	BiCE(K=1)	BiCE(K=10)	aE-SuS	iPIM	iPIM+aPIM	aE-SuS+aPIM
IEEE 14	58	75	2.5	1.4	1.4(2.3*)	1.3(2.8)
IEEE 30	36	11	3.3	3.2	1.7(3.1)	2.4(5.1)
IEEE 57	16	23	9.0	8.2	4.1(8.0)	6.0(12)
IEEE 118	19	18	8.1	6.1	3.6(7.0)	4.0(8.2)
IEEE 300	1.1	0.68	12	12	5.1(10)	5.1(11)

* The number in parenthesis is the relative efficiency without considering the pilot cost.

Table 7.5: Relative efficiency of different sampling-based methods (Scenario 2, $p_f \approx 10^{-5}$).

	BiCE(K=1)	BiCE(K=10)	aE-SuS	iPIM	iPIM+aPIM	aE-SuS+aPIM
IEEE 14	4.8	16	12	1.1	3.5(5.4*)	5.0(9.8)
IEEE 30	45	69	10	11	6.1(11)	6.4(13)
IEEE 57	$5.0 \cdot 10^2$	$5.0 \cdot 10^2$	29	38	15(29)	19(40)
IEEE 118	18	10	35	31	16(31)	18(37)
IEEE 300	6.4	0.67	53	61	35(68)	36(73)

* The number in parenthesis is the relative efficiency without considering the pilot cost.

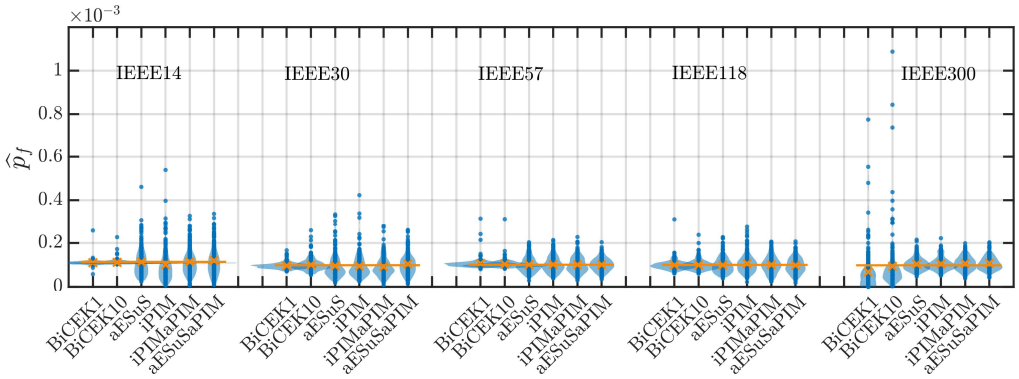


Figure 7.1: The failure probability estimates of different adaptive MCS methods (Scenario 1, $p_f \approx 10^{-4}$).

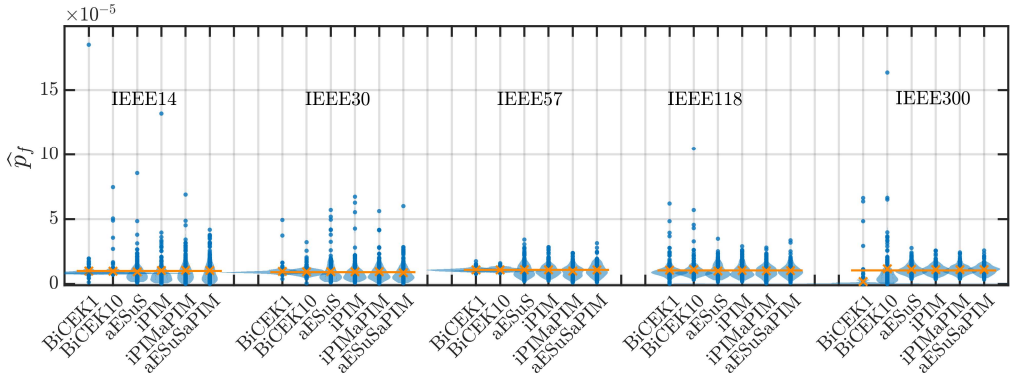


Figure 7.2: The failure probability estimates of different adaptive MCS methods (Scenario 1, $p_f \approx 10^{-4}$).

7.3.4 Discussion of results

Based on the above results, we summarize the strengths and weaknesses of the different methods and discuss the optionality of obtaining an unbiased estimator.

7.3.4.1 The pros and cons

We first discuss the performance of the BiCE methods, which is demonstrated in the first two columns of Tables 7.4 and 7.5. The first column shows the relative efficiency of the BiCE method with a single categorical distribution as the parametric model, while in the second column, we consider a categorical mixture with $K = 10$ components. The mixture model is more flexible but also easier to be overfitted with limited data.

The BiCE methods achieve better results than the multi-level splitting methods in lower-dimensional benchmarks, i.e., in IEEE14, 30, and 57. We attribute this advantage to the independence of the samples generated in the BiCE methods. On the other hand, the BiCE method, whether employing a single or mixture categorical distribution, performs poorly in the IEEE300 benchmark due to the degeneration of the IS weights, which is a well-known issue when performing importance sampling in high dimensions. Specifically, as shown in Fig. 7.1 and 7.2, the BiCE method with single categorical distribution outputs a highly skewed estimator whose bias and variance cannot be accurately estimated through 200 repetitions. Note that although the empirical bias is significant, the BiCE estimator is guaranteed to be unbiased [15], and therefore, the corresponding relative efficiency from 200 repetitions is not trustworthy.

The non-parametric multi-level splitting methods are superior in high-dimensional settings. For instance, in the IEEE300 benchmark, the relative efficiency of aE-SuS (or PiMs) is considerably larger than that of BiCE. Furthermore, the performance of aE-SuS, iPIM, and iPIM+aPIM varies in different settings. In all benchmarks, the iPIM+aPIM method is less efficient than its aE-SuS and iPIM counterparts when taking into account the cost of the pilot run (last four columns of Tables 7.4 and 7.5). We also report the relative efficiency of iPIM+aPIM, calculated without considering the cost of the pilot run, and even in such a case, the relative efficiency of iPIM+aPIM is similar to

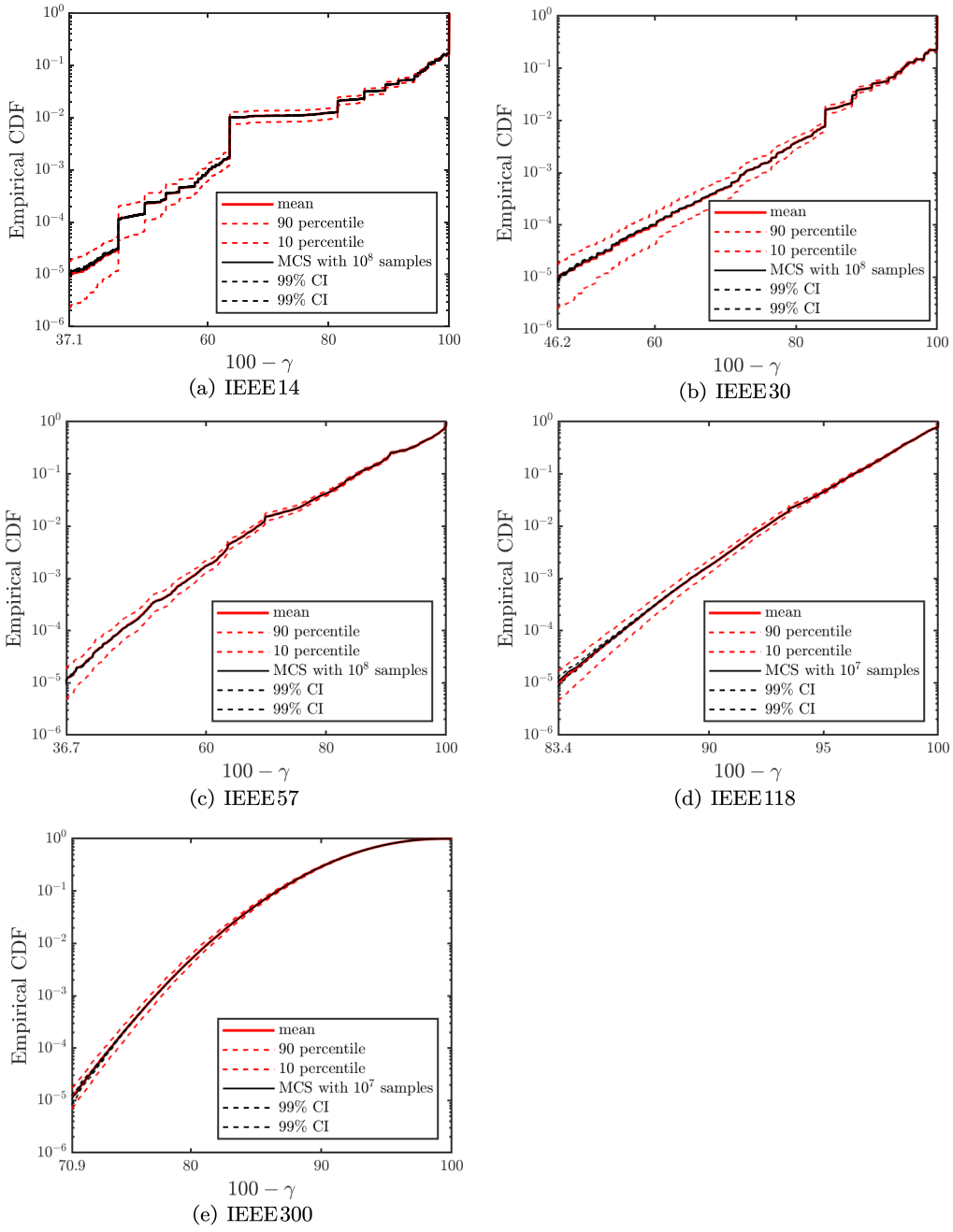


Figure 7.3: Empirical CDF of the survival rate, $100 - PBS(\boldsymbol{X})$, across different benchmarks. The results with aE-SuS are shown in red; the results obtained with MCS with 10^7 samples are in black.

that of aE-SuS. Nevertheless, iPIM+aPIM still significantly outperforms crude Monte Carlo and is better suited when manipulating the intermediate samples is memory-expensive. The performance of iPIM and aE-SuS is similar in all benchmarks except the IEEE14, where we observe significant discontinuity in the CDF of the network performance (See Fig. 7.5). Since the aE-SuS method can handle the big 'jumps' in network performance, it performs better than the iPIM (or the standard SuS) in cases with a small number of components. For larger IEEE benchmarks, the CDF of the network performance becomes smoother, and aE-SuS degenerates into the SuS, which performs similarly to iPIM. The relative efficiency of the two methods is, therefore, similar in these larger benchmarks.

It is also evident that the relative efficiency of the multi-level splitting methods is significantly higher in Scenario 2, where the failure probability is one order of magnitude smaller. However, this is not true for the BiCE methods. As shown in Fig. 7.1 and 7.2, the strong outliers (e.g., in IEEE14 and IEEE118 benchmarks in Scenario 2) lead to a large MSE and, consequently, a low relative efficiency of these estimators.

Overall, the BiCE methods are more efficient in low to moderate dimensions, while aE-SuS and the iPIM are better for addressing high-dimensional problems, whereby both methods are similar to the standard SuS.

7.3.4.2 The optionality of obtaining an unbiased estimator

We use the MSE to assess the accuracy of different adaptive Monte Carlo estimators of the failure probability, $\widehat{p_f}$. The MSE can be decomposed as:

$$\mathrm{MSE}(\widehat{p_f}) = (\mathbb{E}(\widehat{p_f}) - p_f)^2 + \mathrm{Var}(\widehat{p_f}). \tag{7.9}$$

This indicates that for a given rare event estimation problem, the MSE of a failure probability estimator is related to the bias and the variance. An unbiased estimator with a large variance is also prone to give a poor failure probability estimate that is far from the true value. Eq. (7.9) can be rewritten in function of the relative bias $\epsilon \triangleq \frac{\mathbb{E}(\widehat{p_f})-p_f}{p_f}$ and the c.o.v. $\delta \triangleq \sqrt{\frac{\mathrm{Var}((\widehat{p_f}))}{\mathbb{E}^2((\widehat{p_f}))}}$:

$$\mathrm{MSE}(\widehat{p_f}) = p_f^2 \cdot \left(\epsilon^2 + (1+\epsilon)^2 \cdot \delta^2\right). \tag{7.10}$$

In this benchmark study, we estimate the relative bias and c.o.v. for each adaptive Monte Carlo method through 200 independent repetitions of the algorithm. The results are summarized in Table 6-9 in Appendix A. In almost all settings, the c.o.v. of the adaptive Monte Carlo algorithm is considerably larger than its relative bias, and the MSE is dominated by the variance in Eq. (7.9), which can also be seen in Figs. 7.1 and 7.2. Therefore, from the viewpoint of efficiency, as defined in Eq. (7.8)), it is more important to put effort into reducing the variance of the estimator than eliminating its bias.

It should also be stressed that asymptotically unbiased estimators converge to the true failure probability, p_f, as the sample size approaches infinity, so it is advisable to allocate the entire computational budget into a single run in practice. However, this strategy can encounter storage issues when dealing with problems with very high dimensions. In such cases, the average of multiple repetitions of the estimators should be employed, each with a reduced sample size. Note that a smaller sample

size generally results in a larger bias of the estimator, which cannot be mitigated by averaging the results. An unbiased estimator, however, does not suffer from this issue and, hence, is better suited for such an implementation.

7.3.5 A hybrid approach that combines aE-SuS and aPIM

The performance of the aPIM depends highly on the choice of intermediate levels. In [35], the levels are fixed through a pilot run of the iPIM, where the levels are chosen adaptively such that the conditional probability is equal to a constant value p_0. Consequently, the splitting factors s at each level of the aPIM are fixed at $\frac{1}{p_0}$, which is optimal for continuous network performance.

In particular, for smaller networks, there can be a significant discontinuity in the CDF of the network performance, as in the IEEE14 benchmark (see Fig. 7.5(a)). In such a case, selecting levels based on a specified constant p_0 is not just suboptimal but can lead to substantial errors [14]. For the IEEE14 benchmark, the iPIM (and hence also iPIM+aPIM) even gets trapped at the initial level when selecting $p_0 = 0.3$, which cannot be predicted before executing the algorithm.

Conversely, the aE-SuS method adapts the conditional probabilities $p_t, t = 1, \cdots, T$, as well as the sample size per level N, based on the empirical conditional CDF at each level, making it well-suited for managing network performance with substantial discontinuities. However, due to the adaptation of levels, conditional probabilities, and sample size, the aE-SuS estimator can be biased. This motivates the idea of combining the strengths of both aE-SuS and aPIM to produce an unbiased estimator in situations involving substantial discontinuities in network performance. In particular, we employ aE-SuS as the pilot run for fixing the intermediate levels in aPIM. To account for the difference between the importance function $L(\cdot)$ and the network performance function $g(\cdot)$, a transformation of the levels is necessary. For instance, if the importance function $L(\cdot)$ is defined as $\frac{g(\cdot)}{\gamma}$, the levels identified by aE-SuS need to be scaled by γ before being utilized in the aPIM. Besides a different pilot run, we calculate the splitting factor at the t-th level of the aPIM, denoted as $s^{(t)}$, based on the conditional probability estimate of that level from aE-SuS. Specifically, instead of adopting a fixed splitting factor $s = \frac{1}{p_0}$, we select $s^{(t)}$ as $\frac{1}{\widehat{p}_t^{(\text{aE-SuS})}}$. When significant discontinuity exists in the network performance, these conditional probabilities may differ substantially, and consequently, the splitting factor should also vary. The aPIM estimator in Eq. (7.2) becomes:

$$\hat{p}_f^{(\text{aPIM})} = \left(\prod_{t=1}^{T-1} s^{(t)} \right) |\mathcal{X}_T|, \tag{7.11}$$

The relative efficiency of the aE-SuS+aPIM method across different benchmarks and scenarios is also reported in Tables 7.4 and 7.5 to facilitate comparison with other methods. It is evident that aE-SuS+aPIM outperforms iPIM+aPIM (See the last two columns in Table 7.4 and 7.5) and also crude Monte Carlo, but is less efficient than aE-SuS if taking into account the pilot costs. On the other hand, aE-SuS+aPIM provides an unbiased estimator, which may be favorable for applications in which unbiasedness is a priority.

7.4 Demonstration: a 2000-bus synthetic power grid

According to the benchmark study, aE-SuS and iPIM are the two most competitive techniques for addressing high-dimensional problems, where both methods are similar to the standard SuS. Hence, it is adequate to concentrate on just one of them. In this demonstration, we focus on the aE-SuS method for risk assessment of a large-scale synthetic power grid that comprises over 5,000 components. The system failure probability and Birnbaum importance measures are computed using samples from a single run of the aE-SuS algorithm, both as a function of the network performance threshold. In addition, we also briefly present the results obtained using the aE-SuS+aPIM approach.

7.4.1 The ACTIVSg2000 synthetic power grid

A synthetic power grid is an artificial grid with a similar load profile and power generation as the actual grid. It offers a valuable and convenient tool for researchers to analyze the performance of the grids under different scenarios since the network data for the actual grid is usually confidential and inaccessible. The ACTIVSg2000 test case [4] is a synthetic power grid on Texas's footprint, comprising 2,000 buses and 3,206 transmission lines. The network's topology is depicted in Fig. 7.4, where each graph edge is a transmission line. The red and black dots represent the generators and connecting buses, respectively.

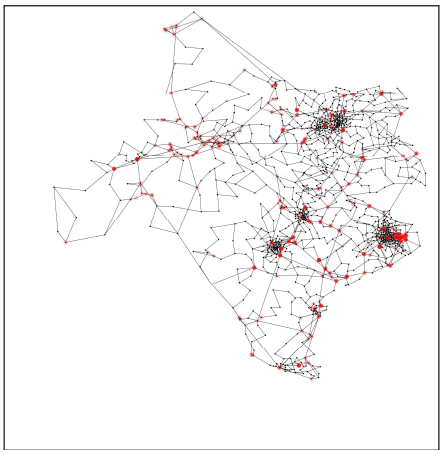


Figure 7.4: The topology of the ACTIVSg2000 synthetic power network. The red and black dots represent the generators and connecting buses, respectively, and the solid lines are transmission lines.

Similarly to the benchmark study, the distribution of the component damage state is summarized in Table 7.3, and we measure the network performance through the $PBS(\boldsymbol{X})$ defined in Eq. (7.7). The empirical CDF of the survival rate, $100 - PBS(\boldsymbol{X})$, is shown in Fig. 7.5, where the mean, ten percentile and 90 percentile of the empirical CDF are calculated through 50 independent repetitions of the algorithm and are compared to crude MCS with 10^7 samples. We further present the c.o.v. square and also the relative efficiency in Fig. 7.6, both as a function of $100 - \gamma$. γ is the threshold in Eq. (7.1). At the initial level (or the MCS level) of the aE-SuS algorithm, the samples are generated independently from the input distribution, and hence, the relative efficiency is approximately equal to 1. In the following conditional levels, sampling is performed through an MCMC algorithm, which

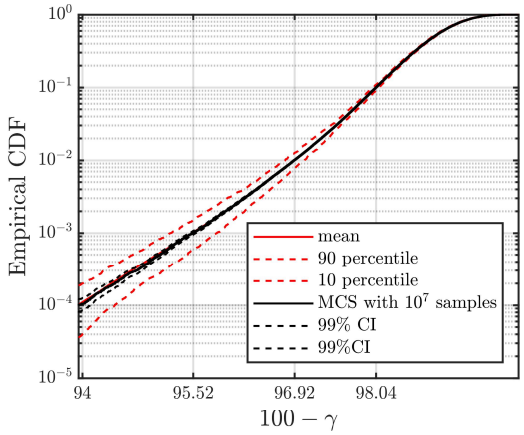


Figure 7.5: The empirical CDF of the survival rate, $100 - PBS(\boldsymbol{X})$. The results with aE-SuS are shown in red; the results obtained with MCS with 10^7 samples are in black. The x-axis tick labels indicate the intermediate levels.

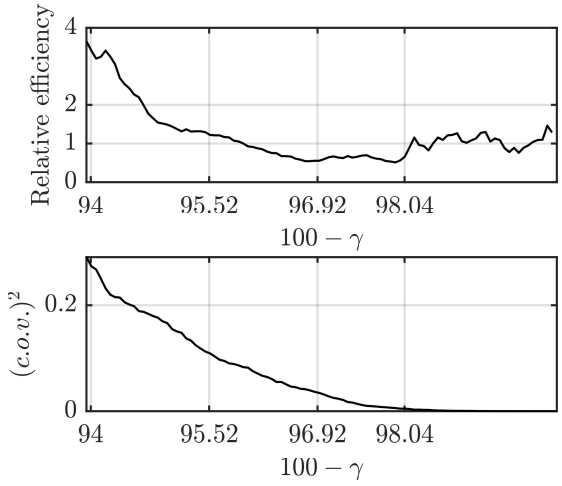


Figure 7.6: The relative efficiency and the $(c.o.v)^2$ of the aE-SuS method. The x-axis tick labels indicate the intermediate levels.

produces dependent samples. Consequently, the relative efficiency drops at the beginning of the second level but increases exponentially thereafter. As the failure probability decreases, the aE-SuS method becomes increasingly more efficient than crude MCS. When $p_f = 10^{-4}$, aE-SuS is around four times more efficient than its MCS counterpart with c.o.v. 0.53 and cost $9,240$. For comparison, when excluding the pilot cost, the relative efficiency for the hybrid method aE-SuS+aPIM is approximately 2.4 with c.o.v. 0.70, and computational cost $8,402$.

7.4.2 Criticality analysis through the aE-SuS algorithm

In network reliability assessment, the criticality of each component is another important aspect of interest. Various reliability-based component importance measures have been developed for quantitatively describing such criticality, which include among others the Birnbaum measure (BM) [5],

improvement potential [41], Fussell-Vesely measure [40, 22], Bayesian importance measure [5]. Initially, most of these measures were proposed for binary components, meaning that the component can only be in either a failure or functional state. However, multiple extensions to accommodate multi-state components can be found [8, 28, 45, 36]. One possible approach to extend the importance measure to multi-state or continuous components is by introducing a threshold at the component level, which differentiates between failure and survival states.

In this demonstration, we adopt the BM as the importance measure and assume that a component fails when it is subjected to at least major damage. Specifically, let $\overline{C_i} \triangleq \{x_i = \text{complete damage}\}$, $F(\gamma) \triangleq \{\boldsymbol{x} : PBS(\boldsymbol{x}) \geq \gamma\}$, and C_i, denote the component failure, system failure, component survival, respectively. Note again that the failure domain $F(\gamma)$ in this demonstration is a function of the threshold γ. The BM measure is then defined as the partial derivative of the network failure probability $\Pr(F(\gamma))$ with respect to the component failure probability, denoted as $\Pr(\overline{C_i})$:

$$BM_i(\gamma) \triangleq \frac{\partial \Pr(F(\gamma))}{\partial \Pr(\overline{C_i})} = Pr(F(\gamma)|\overline{C_i}) - Pr(F(\gamma)|C_i). \tag{7.12}$$

Applying Baye's rule, the equation can be reformulated as [47]:

$$BM_i(\gamma) = \frac{\Pr(\overline{C_i}|F(\gamma)) \cdot \Pr(F(\gamma))}{\Pr(\overline{C_i})} - \frac{(1 - \Pr(\overline{C_i}|F(\gamma))) \cdot \Pr(F(\gamma))}{1 - \Pr(\overline{C_i})}, \tag{7.13}$$

where the component failure probability $\Pr(\overline{C_i})$ and the conditional component failure probability $\Pr(\overline{C_i}|F(\gamma))$ can be estimated using the samples from the first unconditional level and samples conditional on the failure event $F(\gamma)$, respectively. $\Pr(F(\gamma))$ is the main output of the aE-SuS algorithm.

For illustration, we pick four representative thresholds with the target failure probabilities equal to $10^{-1}, 10^{-2}, 10^{-3}$, and 10^{-4}. The BM is then estimated for each threshold through a single run of the aE-SuS algorithm. Fig. 7.7 demonstrates the results for the most important 20 network components, where the black dots represent the value averaged over 50 independent repetitions of the aE-SuS algorithm, and the red crosses depict the MCS reference. It is evident that the aE-SuS estimates agree well with the MCS reference, especially in the first three cases. Note that these sensitivity results can also be adopted as a diagnosis for the MCMC sampler at different levels in aE-SuS.

In the scenario where the failure probability is 10^{-4}, two components exhibit a significantly higher BM value than the remaining components. They are both connecting buses located in the center of the city of Houston (See Fig. 7.8). The failure of these two buses cuts off the associated load and breaks down the transmission lines linked to the buses, leading to a critical load shedding in the network. The remaining critical components are also demonstrated in Fig. 7.8 with blue circles representing the buses or generators and blue links representing the transmission lines.

7.5 Conclusions

The primary focus of this work is on estimating the occurrence probability of large blackouts within power grids. Given that power grid systems are not inherently coherent, and large blackouts are typically rare events, adaptive Monte Carlo methods appear to be particularly suitable. We test

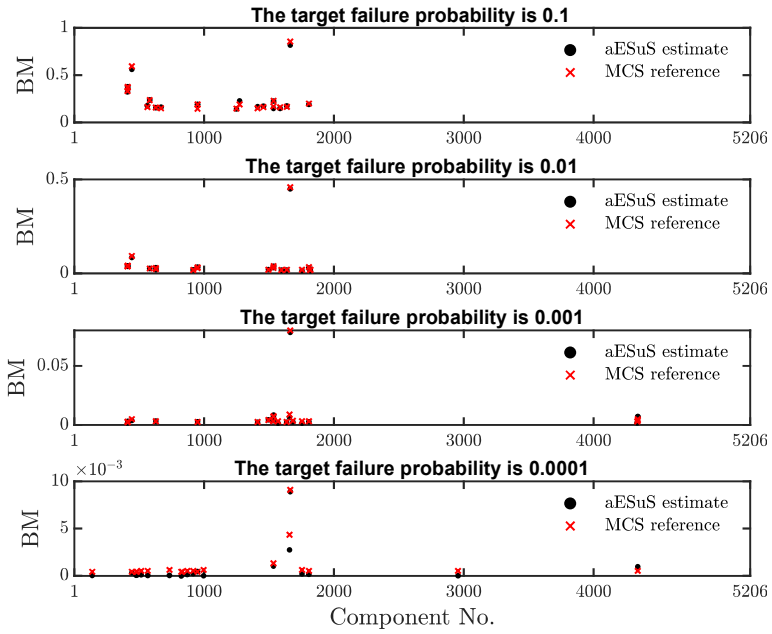


Figure 7.7: The Birnbaum's measure of the most important 20 components. The black dots represent the average value over 50 independent runs of the aE-SuS algorithm, while the red crosses represent the MCS reference.

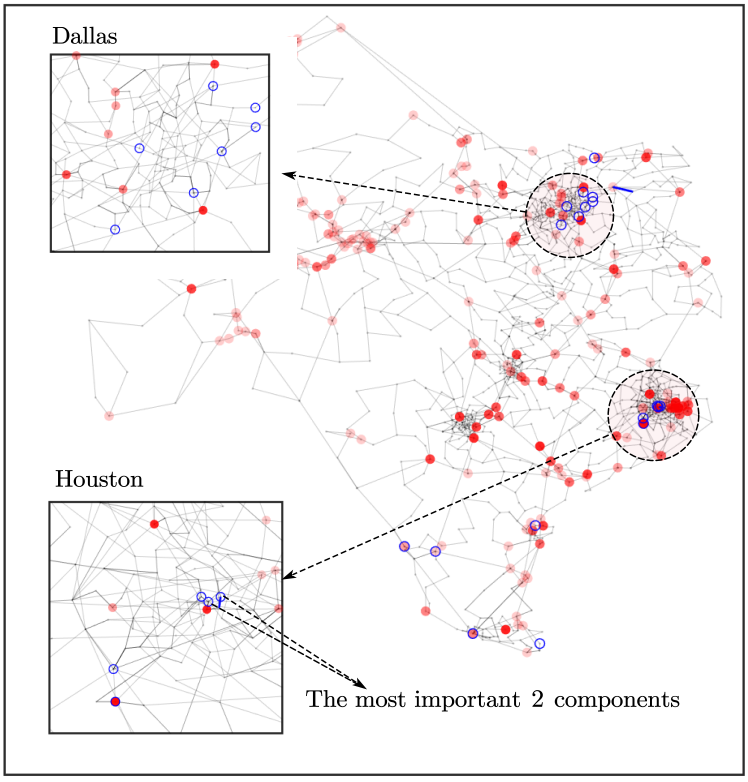


Figure 7.8: The most important 20 components in the ACTIVSg2000 power network. The blue circles represent the critical buses or generators, and the blue link signifies the critical transmission lines.

and compare the performance of four recently developed adaptive Monte Carlo methods, namely the adaptive effort subset simulation (aE-SuS), Bayesian improved cross entropy method (BiCE), and particle integration methods (PIMs), either interacting or annealed, for direct current optimal power flow problems in different IEEE benchmark models. The dimension of the benchmarks ranges from dozens to several hundred. We additionally introduce a new hybrid algorithm that employs aE-SuS as the pilot run of aPIM.

While the unbiased BiCE appears to be the method of choice for smaller systems and provides a significant improvement over crude MCS, there is still room for more efficient methods to handle large dimensional problem settings. The aE-SuS method performs well in large-scale applications where its performance is comparable to iPIM (or SuS). In all benchmarks, the aPIM is less efficient than its aE-SuS and iPIM counterparts due to the additional cost of obtaining appropriate intermediate levels but gives the option to analysts to obtain unbiased estimates if so desired.

We further investigate the performance of aE-SuS for risk assessment of the Texas synthetic power grid, which comprises over 5,000 components. Therein, the failure probability and Birnbaum importance measures are computed using samples from a single run of the aE-SuS algorithm, both as a function of the network performance threshold. The results verify that the aE-SuS method is promising for rare event estimation in large power grids.

Future directions encompass the development of MCMC algorithms tailored for efficient sampling within discrete spaces and the integration of dimension reduction techniques with the BiCE methods. Furthermore, it is of interest to investigate various metrics for assessing the accuracy of sampling-based algorithms, particularly those that provide a more refined description of computational costs.

7.6 Acknowledgements

The first author gratefully acknowledges the financial support of the China Scholarship Council, and the second and fourth authors are grateful for the US National Science Foundation (NSF) grant CMMI-2037545 and the National Institute for Testing and Technology (NIST) award F 70NANB15H044. We thank Prof. Adam Birchfield at Texas A&M University for his support with the Texas Synthetic Grid Data.

7.A Supplementary results for the benchmark study

Table 6: The relative bias of the different methods (Scenario 1, $p_f \approx 10^{-4}$).

	BiCE(K=1)	BiCE(K=10)	aE-SuS	iPIM	iPIM+aPIM	aE-SuS+aPIM
IEEE 14	-0.03 ± 0.01 [1]	-0.02 ± 0.01	-0.01 ± 0.04	-0.11 ± 0.06	0.00 ± 0.04	0.05 ± 0.04
IEEE 30	-0.04 ± 0.01	0.01 ± 0.02	-0.03 ± 0.04	-0.04 ± 0.04	-0.07 ± 0.04	0.06 ± 0.03
IEEE 57	0.05 ± 0.02	0.02 ± 0.01	-0.00 ± 0.02	0.01 ± 0.03	-0.00 ± 0.03	0.00 ± 0.02
IEEE 118	-0.01 ± 0.02	-0.01 ± 0.02	-0.01 ± 0.03	0.06 ± 0.03	0.01 ± 0.03	-0.02 ± 0.03
IEEE 300	-0.33 ± 0.06	-0.07 ± 0.09	0.06 ± 0.02	0.07 ± 0.02	0.08 ± 0.02	0.10 ± 0.02

Table 7: The relative bias of the different methods (Scenario 2, $p_f \approx 10^{-5}$).

	BiCE(K=1)	BiCE(K=10)	aE-SuS	iPIM	iPIM+aPIM	aE-SuS+aPIM
IEEE 14	-0.01 ± 0.09	-0.02 ± 0.05	-0.10 ± 0.06	0.03 ± 0.19	0.05 ± 0.07	0.04 ± 0.06
IEEE 30	0.06 ± 0.03	0.03 ± 0.03	0.05 ± 0.07	0.04 ± 0.07	0.05 ± 0.06	-0.05 ± 0.06
IEEE 57	-0.00 ± 0.01	-0.00 ± 0.01	0.01 ± 0.04	0.00 ± 0.03	-0.09 ± 0.04	-0.00 ± 0.03
IEEE 118	0.02 ± 0.05	0.06 ± 0.06	-0.04 ± 0.03	-0.01 ± 0.04	-0.03 ± 0.04	-0.02 ± 0.03
IEEE 300	-0.83 ± 0.05	0.12 ± 0.26	0.08 ± 0.03	0.05 ± 0.03	0.06 ± 0.02	0.06 ± 0.02

Table 8: The coeff. of variation of the different methods (Scenario 1, $p_f \approx 10^{-4}$).

	BiCE(K=1)	BiCE(K=10)	aE-SuS	iPIM	iPIM+aPIM	aE-SuS+aPIM
IEEE 14	0.11	0.10	0.57	0.74	0.53	0.57
IEEE 30	0.15	0.29	0.58	0.56	0.49	0.47
IEEE 57	0.21	0.18	0.35	0.36	0.36	0.30
IEEE 118	0.23	0.24	0.37	0.40	0.39	0.37
IEEE 300	1.33	1.30	0.29	0.28	0.29	0.29

Table 9: The coeff. of variation of the different methods (Scenario 2, $p_f \approx 10^{-5}$).

	BiCE(K=1)	BiCE(K=10)	aE-SuS	iPIM	iPIM+aPIM	aE-SuS+aPIM
IEEE 14	1.28	0.69	0.88	2.67	0.88	0.86
IEEE 30	0.39	0.35	0.91	0.92	0.79	0.81
IEEE 57	0.12	0.12	0.53	0.47	0.47	0.47
IEEE 118	0.65	0.83	0.51	0.52	0.49	0.48
IEEE 300	4.35	3.34	0.37	0.36	0.33	0.33

Table 10: The average cost of the different methods (Scenario 1, $p_f \approx 10^{-4}$).

	BiCE(K=1)	BiCE(K=10)	aE-SuS	iPIM	iPIM+aPIM	aE-SuS+aPIM
IEEE 14	12,030	12,000	10,658	9,250	22,531(13,281)	18,288(8,878)
IEEE 30	12,740	11,230	9,820	9,270	20,828(11,558)	18,857(9,078)
IEEE 57	11,890	11,890	9,240	9,080	18,656(9,576)	18,886(9,726)
IEEE 118	10,000	10,000	9,200	8,930	18,206(9,276)	17,626(8,466)
IEEE 300	10,250	10,000	9,140	9,030	18,459(9,429)	17,393(8,323)

[1] the standard deviation of the relative bias

Table 11: The average cost of the different methods (Scenario 2, $p_f \approx 10^{-5}$).

	BiCE(K=1)	BiCE(K=10)	aE-SuS	iPIM	iPIM+aPIM	aE-SuS+aPIM
IEEE 14	13,020	13,950	12,430	11,560	33,022(21,462)	25,076(12,734)
IEEE 30	14,050	12,260	11,980	11,370	26,291(14,921)	24,115(12,265)
IEEE 57	12,100	12,360	11,140	11,060	22,744(11,684)	21,679(10,569)
IEEE 118	12,000	11,730	11,320	11,140	23,221(12,081)	22,158(10,868)
IEEE 300	12,150	10,260	11,050	10,890	22,491(11,601)	21,520(10,521)

References

[1] S.-K. Au and J. L. Beck. "Estimation of small failure probabilities in high dimensions by subset simulation". In: *Probabilistic Engineering Mechanics* 16.4 (2001), pp. 263–277.

[2] S.-K. Au and Y. Wang. *Engineering Risk Assessment with Subset Simulation*. John Wiley & Sons, 2014.

[3] R. Billinton and W. Li. *Reliability assessment of electric power systems using Monte Carlo methods*. Springer Science & Business Media, 1994.

[4] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye. "Grid structural characteristics as validation criteria for synthetic networks". In: *IEEE Transactions on Power Systems* 32.4 (2016), pp. 3258–3265.

[5] Z. W. Birnbaum. *On the importance of different components in a multicomponent system*. Tech. rep. Washington Univ Seattle Lab of Statistical Research, 1968.

[6] Z. I. Botev and D. P. Kroese. "Efficient Monte Carlo simulation via the generalized splitting method". In: *Statistics and Computing* 22.1 (2012), pp. 1–16.

[7] Z. I. Botev, P. L'Ecuyer, G. Rubino, R. Simard, and B. Tuffin. "Static network reliability estimation via generalized splitting". In: *INFORMS Journal on Computing* 25.1 (2013), pp. 56–71.

[8] D. A. Butler. "A complete importance ranking for components of binary coherent systems, with extensions to multi-state systems". In: *Naval Research Logistics Quarterly* 26.4 (1979), pp. 565–578.

[9] J.-E. Byun and J. Song. "Generalized matrix-based Bayesian network for multi-state systems". In: *Reliability Engineering & System Safety* 211 (2021), p. 107468.

[10] F. Cadini, G. L. Agliardi, and E. Zio. "Estimation of rare event probabilities in power transmission networks subject to cascading failures". In: *Reliability Engineering & System Safety* 158 (2017), pp. 9–20.

[11] H. Cancela, L. Murray, and G. Rubino. "Efficient estimation of stochastic flow network reliability". In: *IEEE Transactions on Reliability* 68.3 (2019), pp. 954–970.

[12] F. Cérou, P. Del Moral, T. Furon, and A. Guyader. "Sequential Monte Carlo for rare event estimation". In: *Statistics and Computing* 22.3 (2012), pp. 795–808.

[13] F. Cérou, A. Guyader, and M. Rousset. "Adaptive multilevel splitting: Historical perspective and recent results". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.4 (2019), p. 043108.

[14] J. Chan, I. Papaioannou, and D. Straub. "An adaptive subset simulation algorithm for system reliability analysis with discontinuous limit states". In: *Reliability Engineering & System Safety* 225 (2022), p. 108607.

[15] J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method for network reliability assessment". In: *Structural Safety* 103 (2023), p. 102344.

[16] J. Chan, I. Papaioannou, and D. Straub. "Bayesian improved cross entropy method with categorical mixture models". (Under review).

[17] J. Chan, I. Papaioannou, and D. Straub. "Improved cross-entropy-based importance sampling for network reliability assessment". In: *Proceedings of the 13th International Conference on Structural Safety & Reliability*. ICOSSAR. 2022.

[18] J. Chan, R. Paredes, I. Papaioannou, L. Duenas-Osorio, and D. Straub. "A comparative study on adaptive Monte Carlo methods for network reliability assessment". In: *Proceedings of the 14th International Conference on Application of Statistics and Probability in Civil Engineering*. ICASP. 2023.

[19] N. L. Dehghani, S. Zamanian, and A. Shafieezadeh. "Adaptive network reliability analysis: Methodology and applications to power grid". In: *Reliability Engineering & System Safety* 216 (2021), p. 107973.

[20] P. Del Moral. *Feynman-Kac Formulae*. Springer, 2004.

[21] P. Del Moral. *Mean Field Simulation for Monte Carlo Integration*. Chapman and Hall/CRC, 2013.

[22] J. Fussell. "How to hand-calculate system reliability and safety characteristics". In: *IEEE Transactions on Reliability* 24.3 (1975), pp. 169–174.

[23] G. Hardy, C. Lucet, and N. Limnios. "K-terminal network reliability measures with binary decision diagrams". In: *IEEE Transactions on Reliability* 56.3 (2007), pp. 506–515.

[24] M. Huber. "A Bernoulli mean estimate with known relative error distribution". In: *Random Structures and Algorithms* 50 (2017), pp. 173–182.

[25] K.-P. Hui, N. Bean, M. Kraetzl, and D. P. Kroese. "The cross-entropy method for network reliability estimation". In: *Annals of Operations Research* 134.1 (2005), pp. 101–118.

[26] R. A. Jabr. "Robust transmission network expansion planning with uncertain renewable generation and loads". In: *IEEE Transactions on Power Systems* 28.4 (2013), pp. 4558–4567.

[27] H. A. Jensen and D. J. Jerez. "A stochastic framework for reliability and sensitivity analysis of large scale water distribution networks". In: *Reliability Engineering & System Safety* 176 (2018), pp. 80–92.

[28] G. Levitin and A. Lisnianski. "Importance and sensitivity analysis of multi-state systems using the universal generating function method". In: *Reliability Engineering & System Safety* 65.3 (1999), pp. 271–282.

[29] F. Li and R. Bo. "DCOPF-based LMP simulation: Algorithm, comparison with ACOPF, and sensitivity". In: *IEEE Transactions on Power Systems* 22.4 (2007), pp. 1475–1485.

[30] J. Li and J. He. "A recursive decomposition algorithm for network seismic reliability evaluation". In: *Earthquake Engineering & Structural Dynamics* 31.8 (2002), pp. 1525–1539.

[31] I. Papaioannou, W. Betz, K. Zwirglmaier, and D. Straub. "MCMC algorithms for subset simulation". In: *Probabilistic Engineering Mechanics* 41 (2015), pp. 89–103.

[32] I. Papaioannou, S. Geyer, and D. Straub. "Improved cross-entropy-based importance sampling with a flexible mixture model". In: *Reliability Engineering & System Safety* 191 (2019), p. 106564.

[33] R. Paredes, L. Dueñas-Osorio, and I. Hernandez-Fajardo. "Decomposition algorithms for system reliability estimation with applications to interdependent lifeline networks". In: *Earthquake Engineering & Structural Dynamics* 47.13 (2018), pp. 2581–2600.

[34] R. Paredes, L. Dueñas-Osorio, K. S. Meel, and M. Y. Vardi. "Principled network reliability approximation: A counting-based approach". In: *Reliability Engineering & System Safety* 191 (2019), p. 106472.

[35] R. Paredes, H. Talebiyan, and L. Dueñas-Osorio. "Path-dependent reliability and resiliency of critical infrastructure via particle integration methods". In: *Proceedings of the 13th International Conference on Structural Safety & Reliability*. IASSAR. 2022.

[36] J. E. Ramirez-Marquez and D. W. Coit. "Composite importance measures for multi-state systems with multi-state components". In: *IEEE Transactions on Reliability* 54.3 (2005), pp. 517–529.

[37] R. Y. Rubinstein. "Optimization of computer simulation models with rare events". In: *European Journal of Operational Research* 99.1 (1997), pp. 89–112.

[38] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, 2016.

[39] B. Stott, J. Jardim, and O. Alsaç. "DC power flow revisited". In: *IEEE Transactions on Power Systems* 24.3 (2009), pp. 1290–1300.

[40] W. Vesely. "A time-dependent methodology for fault tree evaluation". In: *Nuclear Engineering and Design* 13.2 (1970), pp. 337–360.

[41] W. Vesely, T. Davis, R. Denning, and N. Saltos. *Measures of risk importance and their applications*. Tech. rep. Battelle Columbus Labs., 1983.

[42] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education". In: *IEEE Transactions on Power Systems* 26.1 (2010), pp. 12–19.

[43] E. Zio. *Monte Carlo Simulation: The Method*. Springer, 2013.

[44] E. Zio and N. Pedroni. "Reliability analysis of discrete multi-state systems by means of subset simulation". In: *Proceedings of the 17th ESREL Conference*. 2008, pp. 22–25.

[45] E. Zio and L. Podofillini. "Importance measures of multi-state components in multi-state systems". In: *International Journal of Reliability, Quality and Safety Engineering* 10.03 (2003), pp. 289–310.

[46] K. M. Zuev, S. Wu, and J. L. Beck. "General network reliability problem and its efficient solution by subset simulation". In: *Probabilistic Engineering Mechanics* 40 (2015), pp. 25–35.

[47] K. Zwirglmaier, J. Chan, I. Papaioannou, J. Song, and D. Straub. "Hybrid Bayesian networks for reliability assessment of infrastructure systems". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* (2023).