


RESEARCH ARTICLE

Decomposition algorithms for system reliability estimation with applications to interdependent lifeline networks

Roger Paredes¹  | Leonardo Dueñas-Osorio¹ | Isaac Hernandez-Fajardo²

¹Department of Civil and Environmental Engineering, Rice University, Houston, TX 77005, USA

²Engineering Specialist at INGETEC S.A., Bogotá, Colombia

Correspondence

Roger Paredes, Department of Civil and Environmental Engineering, Rice University, Houston, TX 77005, USA.
Email: roger.paredes@rice.edu

Funding information

U.S. Department of Defense, Grant/Award Number: W911NF-13-1-0340; U.S. National Science Foundation, Grant/Award Number: CMMI-1436845 and CMMI-1541033

Handling Editor: Masayoshi Nakashima

Summary

Reliability and risk assessment of lifeline systems call for efficient methods that integrate hazard and interdependencies. Such methods are computationally challenged when the probabilistic response of systems is tied to multiple events, as performance quantification requires a large catalog of ground motions. Available methods to address this issue use catalog reductions and importance sampling. However, besides comparisons against baseline Monte Carlo trials in select cases, there is no guarantee that such methods will perform or scale well in practice. This paper proposes a new efficient method for reliability assessment of interdependent lifeline systems, termed RAILS, that considers systemic performance and is particularly effective when dealing with large catalogs of events. RAILS uses the state-space partition method to estimate systemic reliability with theoretical bounds and, for the first time, supports cyclic interdependencies among lifeline systems. Recycling computations across an entire seismic catalog with RAILS considerably reduces the number of system performance evaluations in seismic performance studies. Also, when performance estimate bounds are not tight, we adopt an importance and stratified sampling method that in our computational experiments is various orders of magnitude more efficient than crude Monte Carlo. We assess the efficiency of RAILS using synthetic networks and illustrate its application to quantify the seismic risk of realistic yet streamlined systems hypothetically located in the San Francisco Bay Region.

KEYWORDS

algorithms, analytical methods, interdependence, network reliability, risk, state-space partition

1 | INTRODUCTION

Lifeline systems (LSs) are the backbone of modern societies, constantly powering essential activities and supplying for basic needs. However, urban areas are growing denser and integrating with new technologies, which in turn increases the load on LSs and renders them more interconnected. Hence, there is a need for assessing LSs' performance under extreme events, such as earthquakes, while considering their increasing interdependencies. However, such a task is at times computationally infeasible due to the scale of systems and the amount of scenarios to consider for guaranteeing quality of estimates.

1.1 | Seismic risk assessment of lifelines

A key factor in seismic risk assessment is the need to consider intensity maps (IMPs) that are hazard consistent. This usually implies simulating a large number of IMPs that are consistent with ground motion models and earthquake sources (ie, fault and background seismicity). Conventional sequential Monte Carlo simulation (MCS) of earthquake activity requires a large number of simulation years to capture rare events, such as large-magnitude earthquakes of interest in risk assessment. Thus, importance sampling (IS) methods have become popular.¹ In particular, Jayram and Baker² introduced an IS approach that expanded the stratified sampling of source magnitudes³ to the simulation of inter-event and intra-event residuals. Their approach generated hazard IMPs using IS and then damage maps (DMPs) based on component fragilities to assess a transportation network performance metric. More recent work relies on optimization to reduce the set of IMPs and DMPs.^{1,4,5} Nevertheless, some methods require a proxy metric that is not guaranteed to correlate well with performance metrics or they do not exploit the network structure of LSs. In contrast, nonsimulation methods account for network structure as they decompose LSs to estimate their expected performance to arbitrary precision. Such decomposition-based methods are in general independent of local fragilities, so results obtained for one IMP can be “recycled” for other IMPs.⁶

Seismic hazard analysis and loss assessments quantify the probability of exceedance of monetary losses.^{1,4,5} While such an assessment is outside the scope of this paper, we still demonstrate via computational experiments how our methodology (detailed below) can be readily implemented for LSs risk analyses by considering a case study with tenths of thousands of ground motions affecting the loss of interdependent LSs' services. While we only quantify the performance loss annual exceedance probability in LSs, such reduced levels of performance could be associated to monetary loss. Moreover, and in contrast to available methods in seismic risk contexts, we neither sample DMPs nor couple system performance assessment to the generation of hazard scenarios. Our approach is rather modular in the sense that it can be supplied with different hazards if respective fragilities are known for creating a catalog of fragility maps (FMPs), giving novel emphasis on the less studied combinatorial structure of the LSs reliability problem as well as, in principle, facilitating loss estimation provided that monetary losses are associated to the unavailability of lifeline services.

1.2 | Reliability assessment of interdependent lifeline systems

The reliability and performance assessment of LSs continues to be an active area of research and a crucial aspect to consider in the study of resilience.^{7,8} Typically, the scale of systems and complexity of analyses narrow the possibilities for practitioners and researchers to sampling-based methods. However, a major limitation when using simulated samples, is the difficulty of determining the precision of estimations,⁹ or that to do so one needs to conduct large computational experiments or exploit particular problem structures.¹⁰ Furthermore, despite recent advancement,¹¹ machine learning algorithms remain problem- or site-specific. To overcome these limitations, few researchers rely on a combination of analytical and sampling methods that estimate either exactly, within theoretical bounds, or within confidence intervals the reliability of infrastructure systems.

An emerging approach for computing the reliability of LSs is based on the efficient state-space partition (SSP) method.¹² State-space partition techniques decompose the space of possible states of a given system into subsets of feasible and infeasible states (eg, link and cut sets for connectivity problems). In fact, this approach was first due to Doulliez and Jamouille¹³ in the context of multistate systems. Then Alexopoulos corrected the method,¹⁴ and it was further extended and applied to stochastic versions of the shortest path, minimum spanning tree, maximum flow, and other classes of network problems.^{12,15} Furthermore, noting that flow problems can be reduced to connectivity problems, the method introduced in the work by Dotson and Gobien¹⁶ is a special case of multistate systems when reduced to binary systems for connectivity analysis. We make this observation given that Dotson and Gobien's work is the basis for methods that were more recently developed in the reliability and earthquake engineering community (eg, Li and He¹⁷ and Liu et al¹⁸). In this paper, we show how the SSP method more comprehensively encompasses many methods that were developed independently, and we improve and augment the capabilities of these methods for interdependent infrastructure system analyses.

Furthermore, one of the first attempts to study interdependent LSs using nonsimulation methods was conducted by Kim, Kang, and Song.⁶ However, the previous work does not consider the bidirectional and cyclic case of interdependencies among systems, which is challenging and essential to shed light on the vulnerability of interdependent LSs.

This paper presents a hazard-independent framework for risk assessment of lifelines that supports interdependencies among systems. It requires hazard and fragility information to be synthesized into a catalog of FMPs. While this work

uses a high-level integration of methodologies for risk assessment of interdependent networks, we improve or extend methodologically each of the base methods here adopted. The core of our contribution can be summarized as follows:

- We provide a comprehensive review of decomposition methods that were developed independently for solving network reliability problems, integrate them under the umbrella of the SSP method, and improve upon them.
- We introduce a new framework for general interdependent network reliability problems that responds to calls for systemic performance assessment¹⁹ and that we term *reliability assessment of interdependent lifeline systems* (RAILS).
- We introduce new SSP algorithms poised for solving RAILS problems efficiently and showcase how RAILS can be used for risk assessment when considering many FMPs (such as in seismic risk) using an approach that recycles computations.

The remainder of this paper is structured as follows. Section 2 describes the estimation of seismic risk in LSs. Section 3 reviews the SSP method for network reliability and introduces a new framework for RAILS problems with efficient algorithms for solving them. Section 4 presents computational experiments including a case study of seismic risk in test networks to showcase the applicability of the new approach. Section 5 offers conclusions from this study and outlines future research directions.

2 | SEISMIC RISK ASSESSMENT

While most network reliability methods model component stochastic capacity using a discrete random variable with associated *probability mass function* (pmf) that is system-homogeneous or system-heterogeneous (ie, components are allowed to have different pmfs), we consider a more general case. We explicitly acknowledge that many hazard events can be verified, and while the set of possible component states remain the same, the pmf associated to such component states is allowed to have a different distributions under each hazard event. Such generalization is key to extend our reliability-based framework to the risk estimation setting in an efficient manner. Herein, an ensemble of components' pmfs under a single hazard event is termed FMP. Most of previous work addresses network reliability in the single FMP setting. On the other hand, most of previous work in risk assessment treats each FMP independently and focuses on reducing the number of FMPs to be considered. In contrast, we seek no reduction but rather “recycle” the information we gain for every given FMP across the whole catalog of FMPs, while preserving rigorous theoretical bounds. This section serves as an introduction to a common approach to assess seismic risk and expected performance of LSs that results into a large number of FMPs to be considered.

The general approach for seismic risk assessment consists of adopting a source model with characteristic information for seismic events of interest, adopting a ground motion model, linking shaking intensities to damage states of structural components with fragility curves, and finally quantifying system performance or eventually losses. The remainder of this section expands on each of the risk assessment steps and discusses approaches in the literature to cope with the large number of FMPs. Also, we describe a new approach that leverages efficient reliability estimation methods that can recycle computations across the set of FMPs.

Seismic source model and catalog of earthquake scenarios (ESs). To conduct a seismic risk assessment on distributed infrastructure, we first need a stochastic earthquake event catalog. We require a description of potential ESs that includes their associated source (eg, fault or area), magnitude, and annual event rate ν .

Ground motion models and IMPs. For each ES, we can associate a ground motion model as follows:¹

$$\ln(Y_z) = \ln(\bar{Y}_z) + \sigma_z \varepsilon_z + \tau_z \eta_z, \quad (1)$$

where Y_z is the seismic intensity measure of interest (eg, spectral acceleration) generated by a known earthquake magnitude at a site z localized at a known distance from the event source. The first term is associated to the median value of the seismic intensity, and it is computed using the seismogenic information of the associated source model. The second and third terms are the intra-event and inter-event residuals, which are usually drawn from multivariate and univariate distributions, respectively. Furthermore, in this paper, we will call IMP a realization of Equation 1 for all sites of interest, and we will consider a constant number, n , of IMPs for each ES, ES_i , in the stochastic earthquake event catalog as in the work by Miller and Baker.⁴ Therefore, the annual rate of each IMP _{ij} , such that $j = 1, \dots, n$ and associated to ES_i , is $\omega_{ij} = \frac{\nu_i}{n}$.

Fragilities and DMPs. Before estimating the system response to a seismic event, it is a common practice to estimate local physical responses and then assess residual service at the system level. More specifically, fragility curves provide an estimate of the probability $\Pr[DS|Y_z]$ of a component at site z reaching a damage state DS given a seismic intensity level

Y_z . The combined knowledge of site intensities from IMPs and element fragilities enables sampling of damaged states for every component of the system, hence called DMPs. These maps further enable evaluation of system-level performance for each DMP realization. In this study, we will not generate DMPs, but rather treat the probabilistic space of the hazard and of the system independently. Thus, we do not miss important configurations simulating DMPs, and our focus is on efficiently handling the combinatorial space of stochastic system configurations.

As a summary, the conventional system risk assessment paradigm follows a sequential approach in which, for each ES, inter- and intra-event residuals are sampled to generate a set of IMPs. Then, failure probabilities are estimated for each system component using their site intensity measures from each IMP and fragility curves. Moreover, for each IMP and their associated element fragilities, a set of DMPs is generated. Finally, the risk assessment is rounded up with a system performance metric computation for each DMP. Figure 1 summarizes the nested structure of the traditional approach, where k , l , and m denote the total number of ESs, the total number of IMPs for each ES, and the total number of DMPs for each IMP, respectively. This approach requires a large amount of system performance evaluations that may not be executable in practice. For this reason, many researchers have proposed optimization approaches that select a subset of maps that preserve site conditions and are consistent with surrogate system performance metrics.^{1,4,5} However, our first observation is that, when sampling, coupling the hazard space to the system space is unnecessary. Moreover, unless a full enumeration of system states is performed before optimizing the selection of DMPs, the system space is not fully available for selection of meaningful system states due to sampling limitations. Note that the number of distinct DMPs grows exponentially in the size of the system. For instance, a system with 64 binary elements has $2^{64} \approx 1.8 \times 10^{19}$ distinct states.

In this paper, we consider stochastic multicommodity flow problems casted as a mixed-integer linear program to measure interdependent network system performance while making system performance computations affordable. Also, we take a considerably different approach for handling large seismic catalogs with respect to existing methods in the sense that we decouple sampling of the hazard space from the system space. More specifically, we decompose the set of possible LSs states into feasible, infeasible, and unexplored subsets. The goal is to do so in such a way that explored subsets (feasible and infeasible ones) account for most of the probability description of the hazard when transformed into element fragilities (this also implies obtaining lower and upper bounded estimates of probabilistic system performance that are tight). The decomposition approach we use is inherently computationally hard, meaning that full decompositions are only attainable for relatively small systems; however, the algorithms we introduce here aim to produce tight bounds within a few iterations of such algorithms. Furthermore, if slow convergence of bounds is unavoidable, we adopt an *importance and stratified sampling* scheme on the remaining unexplored sets that has proven far more efficient than MCS.¹² Our more direct and decoupled treatment of the system's probabilistic state space with respect to previous studies allows us to recycle all computations throughout the seismic catalog as well. The next section expands these concepts and reviews the underlying mathematical principles.

3 | RELIABILITY ASSESSMENT OF INTERDEPENDENT LIFELINE SYSTEMS

We begin this section with preliminaries on network reliability, including the notation that will be used throughout this section. Next, we review the SSP method for single system reliability problems and show notable applications as well as demonstrate how the SSP method comprehensively encompasses decomposition methods in literature. Along the way, we introduce improvements and extensions to these methods by leveraging general principles of the SSP method. Then we introduce a new framework for reliability problems that extends single system reliability to the *system-of-systems* (SOS) setting and that we term RAILS. Moreover, we introduce new SSP algorithms poised to solve RAILS problems efficiently, while considering interdependency constraints and enabling rapid estimates of systemic reliability with rigorous bounds. Finally, we describe model capabilities of RAILS and its application for risk assessment.

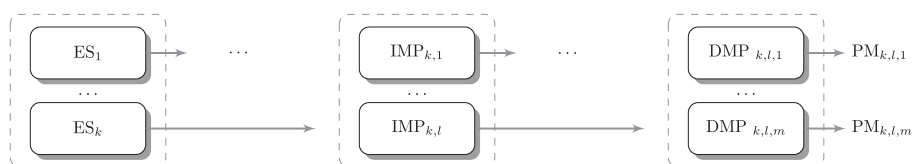


FIGURE 1 Prevaling seismic risk assessment approach. ES, earthquake scenario; IMP, intensity map; DMP, damage map; PM, performance metric

3.1 | Network reliability

Consider a stochastic infrastructure system k represented by a graph $G_k(V_k, E_k)$. Assume an arbitrary labeling of components $\mathcal{L} = \{1, \dots, a\}$ such that $|V_k| + |E_k| = a$. Herein, we will refer to an specific component by its label $i \in \mathcal{L}$, and whenever its distinction from node or link becomes important, we will use one-to-one mappings $\psi : V_k \mapsto \mathcal{L}$ and $\phi : E_k \mapsto \mathcal{L}$, respectively. We assume that component $i \in \mathcal{L}$ can vary its capacity level u_i as a discrete random variable, taking values from the set $\{x_i(1) < \dots < x_i(l_i)\}$ with respective probabilities $\{p_i(1), \dots, p_i(l_i)\}$ (its pmf), where l_i is the number of capacity levels that component $i \in \mathcal{L}$ can take. For example, in a transportation network, a component's capacity level can model the number of available lanes in a bridge crossing. Furthermore, we can use a system-state vector representation based on component capacity levels $X(v) = (x_1(v_1), \dots, x_a(v_a))$ or based on indices $v = (v_1, \dots, v_a)$. Then the state-space Ω of the stochastic system, such as a practical lifeline network, is defined using the Cartesian product of sets as follows:

$$\Omega = \bigotimes_{i=1}^a \{1, \dots, l_i\} = \{(v_1, \dots, v_a) : 1 \leq v_i \leq l_i, \forall i \in \mathcal{L}\}. \quad (2)$$

It is important to realize that the state-space Ω (Equation 2) has a hyper-rectangular structure of dimension $a = |\mathcal{L}|$. Here, index $v_i = 1$ stands for component's $i \in \mathcal{L}$ lowest capacity level (eg, complete damage) and $v_i = l_i$ for its maximum capacity level (eg, no damage). Thus, vertices $v = (1, \dots, 1)$ and $v = (l_1, \dots, l_a)$ are, respectively, the most unfavorable and favorable states in terms of system performance. Furthermore, for infrastructure k , we can adopt an appropriate performance metric¹⁹ in the form of utility function $u_k(v)$ to quantify its level of service. In practice, $u_k(v)$ can be the number of connected terminals in a telecommunication network, the percentage of satisfied customers by a power or water distribution system, or an inverse relation to travel times in a transportation network. Given a prescribed performance threshold D_k , we can define the *infeasible* (\mathcal{I}) and *feasible* (\mathcal{F}) domains $\mathcal{I} = \{v \in \Omega : u_k(v) < D_k(v)\}$ and $\mathcal{F} = \{v \in \Omega : u_k(v) \geq D_k(v)\}$, respectively.

The state-space definitions allow us to clearly define *network reliability*, denoted r , as the probability that system G_k will perform at least at level D_k as follows:

$$r = \Pr[v \in \mathcal{F}] = \sum_{v \in \Omega} \Pr[v] I_{\mathcal{F}}(v), \quad (3)$$

where $I_{\mathcal{F}}$ is the indicator function that outputs 1 if v is in the feasible domain \mathcal{F} and 0 otherwise. Since the size of Ω grows exponentially with the number of unreliable components, computing r from Equation 3 is unattainable even for medium-size networks. Instead, one can use decomposition methods that estimate r in a more compact form.

3.2 | The SSP method

The SSP method exploits the structure of Ω to decompose it into disjoint hyper-rectangular subsets $I_i \subseteq \mathcal{I}$ and $F_i \subseteq \mathcal{F}$ recursively. Let us call \mathcal{I} and \mathcal{F} the lists of disjoint infeasible subsets $I_i \subseteq \mathcal{I}$ and feasible subsets $F_i \subseteq \mathcal{F}$ that are obtained, respectively, after decomposing Ω using an SSP algorithm. Also, any hyper-rectangular subset $S \subseteq \Omega$ can be described by 2 states $\alpha(S) = (\alpha_1, \dots, \alpha_a)$ and $\beta(S) = (\beta_1, \dots, \beta_a)$ such that $S = \{v \in \Omega : \alpha_i \leq v_i \leq \beta_i, \forall i \in \mathcal{L}\}$. We may use the notation $S = [\alpha, \beta]$ to refer to the hyper-rectangular subset S with lowest and highest capacity levels α and β , respectively. We can compute the probability of $v \in S$ as¹²

$$\Pr[v \in S] = \prod_{i \in \mathcal{L}} \sum_{j=\alpha_i}^{\beta_i} p_i(j), \quad (4)$$

where $p_i(j)$ are component's $i \in \mathcal{L}$ discrete state probabilities (pmf) as described previously. Note that for valid pmfs, Equation 4 yields $\Pr[v \in \Omega] = 1$. Using lists of disjoint subsets \mathcal{F} (alternatively \mathcal{I}) and Equation 4, we can compute r as follows:

$$r = \sum_{F_i \in \mathcal{F}} \Pr[v \in F_i] = 1 - \sum_{I_i \in \mathcal{I}} \Pr[v \in I_i]. \quad (5)$$

When estimating r , the efficiency of SSP algorithms depends heavily on the size of lists \mathcal{F} or \mathcal{I} . However, an attractive feature of the SSP method is that it can be implemented for *anytime* algorithms that provide tighter upper and lower bounds on r as they continue running. In general, an SSP algorithm begins with an empty list of feasible subsets $\mathcal{F} = []$, an empty list of infeasible subsets $\mathcal{I} = []$, and a list of unexplored subsets $\mathcal{U} = [\Omega]$. Then it performs the following decomposition to each unexplored set U_{i-1} removed from \mathcal{U} :

$$U_{i-1} = I_i^* \cup F_i^* \cup U_i^*, \quad (6)$$

where $I_i^* \subseteq \mathcal{I}$, $F_i^* \subseteq \mathcal{F}$, and $U_i^* = U_i \setminus (I_i^* \cup F_i^*)$. At iteration $i - 1$, some sets in Equation 6 can result in empty sets, and the superscripts “*” are used to denote that sets may or may not be hyper-rectangular. Then derived sets are stored into lists I , F , and U . However, it is important that before we move any disjoint set obtained from Equation 6 into lists I , F , and U , we make such subsets hyper-rectangular by further partitioning them if not into that form already. A crucial practical issue is that, when partitioning U_i^* into hyper-rectangular subsets U_i , we do not generate too many subproblems U_i to be moved to U , since $|U|$ is always a lower bound on the remaining number of iterations for full decomposition of Ω . Previous research has devised ways to keep the number of subproblems generated up to a or less. Additionally, the way in which SSP algorithms store and remove sets $U_i \in U$ for further decomposition has major impact on their efficiency (we will cover this subsequently). From partial decompositions of Ω , the following bounds hold

$$\sum_{F_i \in F} \Pr[v \in F_i] \leq r \leq 1 - \sum_{I_i \in I} \Pr[v \in I_i]. \quad (7)$$

We denote P_U the gap between lower and upper bounds in the inequality above. State-space partition algorithms have been implemented for many problems such as stochastic shortest path problems,²⁰ multiterminal and multicommodity flow problems,²¹ minimum spanning trees, and other classes of network flow problems.¹⁵ Regardless of the specific problem, SSP algorithms exploit the problem structure of *coherent* systems to classify entire hyper-rectangular subsets of Ω based on the knowledge of one feasible/infeasible state. Coherent systems exhibit a useful property in that if we select any 2 states $v', v \in \Omega$ such that $v'_i \geq v_i$ for all $i \in \mathcal{L}$, ie, system-state v' has components with capacity levels greater or equal to state v , then it always holds that $u_k(v) \leq u_k(v')$. In other words, $u_k(\cdot)$ is monotonic with respect to component reliabilities.²² Intuitively, this means that if a subset of components is failed, yet the system as a whole is safe, then any state with same nonfailed components but less failures is also safe. A symmetric argument can be drawn for an unsafe system with a subset of nonfailed components. Next, we review 2 important SSP algorithm strategies.¹⁵

Feasible-based SSP (F-SSP): The first approach aims at deriving a large feasible set at every iteration of Equation 6. We begin exploring $U_0 = \Omega$ from a state that we know is feasible, say $v = \beta(U_0)$. Note that if v is infeasible, from coherence of $u_k(\cdot)$, the whole state-space is infeasible and it is trivial to compute $r = 0$. Starting from v , we use a problem-specific subroutine **feasible_v0**(v, α, β) to derive a deeper state $v^0 \leq v$ that is also feasible. We defer the specific implementation of **feasible_v0**(\cdot) for particular problems covered later in this section. Using this procedure at any iteration $i - 1$ of Equation 6 and provided that v is feasible, we can derive a disjoint hyper-rectangular subset $F_i^* = [v^0, v] = F_i$ that contains feasible states and with extreme vertices $\alpha(F_i) = v^0$ and $\beta(F_i) = v$. Also, we let I_i^* be the empty set and $U_i^* = U_{i-1} \setminus F_i$. In particular, U_i^* is not guaranteed to be hyper-rectangular, and thus, we need to further partition it into disjoint hyper-rectangular sets. We can partition U_i^* into disjoint hyper-rectangular subsets $U_{(i,j)}$ as follows:¹²

$$\begin{aligned} U_{(i,j)} = \{v \in \Omega : & v_q^0 \leq v_q \leq \beta_q \text{ for } q < j, \\ & \alpha_j \leq v_j < v_j^0 \text{ for } q = j, \\ & \alpha_q \leq v_q \leq \beta_q \text{ for } q > j\}, j \in \mathcal{L}. \end{aligned} \quad (8)$$

For some advanced iteration $i - 1 > 0$, we may find that $\beta(U_{i-1})$ is infeasible. In such cases, we let $I_i^* = I_i = U_{i-1}$, and let remaining subsets in Equation 6 be empty sets. At the end of each iteration, we move all derived nonempty sets F_i , I_i , and $U_{(i,j)}$ for all $j \in \mathcal{L}$ to respective lists F , I , and U . Each iteration proceeds removing a subset U_{i-1} from U until there are no unexplored subsets left or until the bounds on r yield enough precision. Algorithm 1 outlines steps for a full decomposition of Ω using a feasible-based SSP strategy with the possibility of early termination should P_U be less than the target gap tolerance.

Infeasible-based SSP (I-SSP): This approach proceeds symmetrically to the F-SSP; however, we outline the main steps here for completeness. At any iteration $i - 1$, we begin at state $v = \alpha(U_{i-1})$. If $v = \alpha(U_{i-1})$ is feasible, we let $F_i^* = F_i = U_{i-1}$, and let remaining subsets in Equation 6 be empty sets (trivial case). Otherwise, we use a problem-specific routine **infeasible_v0**(v, α, β) to derive a deeper state $v^0 \geq v$ that is also infeasible. Then we let $I_i^* = I_i = [v, v^0]$, and $U_i^* = U_{i-1} \setminus I_i$ is decomposed into hyper-rectangular subsets using the analog to Equation 8 as follows:

$$\begin{aligned} U_{(i,j)} = \{v \in \Omega : & \alpha_q \leq v_q \leq v_q^0 \text{ for } q < j, \\ & v_j^0 < v_j \leq \beta_j \text{ for } q = j, \\ & \alpha_q \leq v_q \leq \beta_q \text{ for } q > j\}, j \in \mathcal{L}. \end{aligned} \quad (9)$$

Nonempty subsets $U_{(i,j)}$ are moved to U . We extract a new subproblem from U and repeat this process until U is empty or until P_U is small enough. Algorithm 2 outlines the steps for fully or partially decomposing Ω using an infeasible-based SSP.

Implementation remarks: Any subset $S \subseteq \Omega$ can be stored with only 2 vectors of indices, namely, $\alpha(S)$ and $\beta(S)$. However, for very large networks, storing all disjoint subsets $F_i \in F$ and $I_i \in I$ in the main memory is not practical. If storage in external memory is not possible, one can simply store lower and upper bounds. In particular, one can initialize lower bound $P_F = 0$ and upper bound $P_I = 1$ and replace steps 9 and 14 in Algorithms 1 and 2 with corresponding update rules $P_F \leftarrow P_F + \Pr[v \in F_i]$ and $P_I \leftarrow P_I - \Pr[v \in I_i]$.¹² However, when considering many FMPs, discarding subsets in F and I can result into a considerable amount of recomputation.

A crucial issue when implementing SSP algorithms concerns appropriate handling of U , as in certain cases, it can result into $|U|$ that is intractably large. It has been pointed out that SSP algorithms work in a branch-and-bound fashion.^{12,23} Therefore, a way to better describe the previous issue is to use a tree-like representation among unexplored subsets. For example, when removing the first input subset U_{i-1} from U and decomposing it (Equation 6), we expect to generate at most a new subproblems $U_{(i,j)}$ (Equations 8 and 9) to be *pushed* into U . In fact, numerical experiments have shown that a depth-first-search (DFS) strategy keeps $|U|$ small. On the other hand, prioritizing the exploration of subproblems with largest probability $\Pr[v \in U_i]$ will shrink bounds in Equation 7 faster, but potentially incur into large $|U|$ during the decomposition process. In practice, a DFS strategy is implemented with a last-in-first-out (LIFO) policy when handling U , whereas prioritizing exploration of $U_i \in U$ with largest probability is implemented with a Heap data structure.²⁰ Moreover, the literature shows another strategy based on recursion in the context of s - t network reliability.¹⁷ While this recursion is essentially a DFS strategy (explore first child problem first) that yields slow convergence of Equation 7, it does a slightly worst job at keeping $|U|$ small than using a simple LIFO policy, as shown in our numerical experiments; we will refer to this strategy as Recursion. Additionally, prioritizing the removal of subset $U_i \in U$ with largest probability was independently suggested by other researchers.²⁴ The *extract* and *push* steps in Algorithms 1 and 2 will assume a preset strategy for handling U (namely, LIFO, Recursion, or Heap).

Also, Lim and Song²⁴ introduced a new technology to speed up convergence of Equation 7. In the context of s - t network reliability and using an F-SSP algorithm with Heap handling of U , Lim and Song proposed finding a deeper state v^0 that maximizes the reliability of the path connecting s and t , which in turn yields $F_i = [v^0, \beta]$ that shrinks P_U faster. To find the *most reliable path*, they introduce a modified Dijkstra's algorithm. However, later in this section, we will show a generalization of this approach that needs no modified algorithms and applies to multistate systems.

Algorithm 1 Feasible-based SSP

```

1: procedure F-SSP( $F, I, U, \Omega, \text{gap}$ )
2:    $i \leftarrow 0$ 
3:   while  $P_U > \text{gap}$  do
4:      $i \leftarrow i + 1$ 
5:     Extract subset  $U_{[i]}$  from  $U$ 
6:     Let  $[\alpha, \beta] = U_{[i]}$ 
7:     if  $\beta$  is infeasible then
8:       Let  $I_i = [\alpha, \beta]$ 
9:       Move  $I_i$  to list  $I$ 
10:    else
11:       $v \leftarrow \beta$ 
12:       $v^0 \leftarrow \text{feasible}_v0(v, \alpha, \beta)$ 
13:       $F_i \leftarrow [v^0, v]$ 
14:      Move  $F_i$  to list  $F$ 
15:      Derive  $U_{(i,j)}$ 
16:      Push any  $U_{(i,j)} \neq \emptyset$  into  $U$ 
17:    end if
18:  end while
19: end procedure

```

▷ P_U from Equation 7

▷ from Equation (8)

Algorithm 2 Infeasible-based SSP

```

1: procedure I-SSP( $F, I, U, \Omega, \text{gap}$ )
2:    $i \leftarrow 0$ 
3:   while  $P_U > \text{gap}$  do ▷  $P_U$  from Equation 7
4:      $i \leftarrow i + 1$ 
5:     Extract subset  $U_{i-1}$  from  $U$ 
6:     Let  $[\alpha, \beta] = U_{i-1}$ 
7:     if  $\alpha$  is feasible then
8:       Let  $F_i = [\alpha, \beta]$ 
9:       Move  $F_i$  to list  $F$ 
10:    else
11:       $v \leftarrow \alpha$ 
12:       $v^0 \leftarrow \text{infeasible\_v0}(v, \alpha, \beta)$ 
13:       $I_i \leftarrow [v, v^0]$ 
14:      Move  $I_i$  to list  $I$ 
15:      Derive  $U_{(i,j)}$  ▷ from Equation (9)
16:      Push any  $U_{(i,j)} \neq \emptyset$  into  $U$ 
17:    end if
18:  end while
19: end procedure

```

Lastly, any undirected link $(i, j) \in E_k$ can be handled in our framework by augmenting E_k to contain both (i, j) and (j, i) such that $i, j \in V_k$ and substituting any appearance of ϕ by a many-to-one mapping ϕ' such that $\phi'(i, j) = \phi'(j, i)$, ie, links (i, j) and (j, i) have the same label. Note that when there are undirected edges, we update our definition of a , the number of components, to $a = |\mathcal{L}|$ such that reversed edges in E_k sharing labels are counted once.

3.3 | Notable network reliability problems

This subsection illustrates the SSP method in the context of classic network reliability problems. These problems have been used for quantifying probabilistic performance of LSs, and before extending such analyses to our SOS case, we show how we can formulate classical network reliability problems in a generalized form using the SSP method. We will make connections in literature that have been overlooked to date and show example applications. However, to make this exposition more amenable, we will introduce the following notation for any hyper-rectangular subset $S \subseteq \Omega$. In particular, for a hyper-rectangular subset of Ω defined as $S = \{v \in \Omega : \alpha_i \leq v_i \leq \beta_i, \forall i \in \mathcal{L}\}$ where $\alpha, \beta \in \Omega$ are the farthest diagonal vertices in S such that $\alpha \leq \beta$, we will write $S = \Omega[\alpha_1 : \beta_1, \dots, \alpha_a : \beta_a]$ with $\alpha_i : \beta_i$ standing for the discrete interval $[\alpha_i, \beta_i] = \{v_i \in \{1, \dots, l_i\} : \alpha_i \leq v_i \leq \beta_i\}$ in dimension $i \in \mathcal{L}$. Additionally, we consider the following convention for short-hand notation of entries $\alpha_i : \beta_i$. If $\alpha_i = 1$, then we will omit α_i . Similarly, if $\beta_i = l_i$, then we will omit β_i . On the other hand, if $\alpha_i = v_i = \beta_i$, then we may substitute $\alpha_i : \beta_i$ by simply v_i . We will exemplify this notation after its first usage below.

Source-terminal network reliability: This problem has been proven to be #P-complete,²⁵ and it concerns computing the probability of finding a path between nodes s and t such that $s, t \in V_k$ given that links can fail with known probability p_e . To adapt the exposition of Section 3.1, we assume that every link $(e_a, e_b) \in E_k$ labeled $i = \phi(e_a, e_b)$ can fail and that it has capacity $u_i \in \{x_i(1) < x_i(2)\} = \{0 < 1\}$ with respective probabilities $\{p_i(1), p_i(2)\} = \{p_e, 1 - p_e\}$, ie, $\Pr[x_i(1)] = p_e$ and $\Pr[x_i(2)] = 1 - p_e$ are its capacity pmf. Also, \mathcal{L} keeps labels for links only, given that nodes are perfectly reliable.

SSP algorithms find paths and cuts between s and t and decompose Ω into disjoint sets that are known to contain s - t -connected (feasible) and s - t -disconnected (infeasible) states. For example, consider the 4-node network with 5 links in Figure 2A. Let us arbitrarily choose an F-SSP strategy with a Recursion handling of U to compute r . Since only links can fail, we consider labeling $\mathcal{L} = \{1, 2, 3, 4, 5\}$ as shown in Figure 2A. Remember that at the beginning, F and I are empty, and $U = [\Omega]$. For the first iteration $i = 1$ (Algorithm 1), we extract the *last* subset in U and let α and β be the lowest and highest capacity levels, respectively, within $U_0 = \Omega = \{1, 2\}^5$. Remember that $v \in \Omega$ is a vector of indices, where $v_1 = 1$ means that link labeled 1 has capacity $u_1 = x_1(1) = 0$ (ie, is failed) and $v_1 = 2$ means that capacity $u_1 = x_1(2) = 1$ (ie, is not failed). After verifying that $\beta = (2, 2, 2, 2, 2)$ is feasible, ie, there is a path between nodes s and t when all links are not failed, we set $v = \beta$. Let us show how to find a deeper state v^0 using subroutine **feasible_v0**(v, α, β) for the case of s - t connectivity. A shortest path sequence at state v can be obtained using Dijkstra's algorithm while assuming equal link weights. Consider the shortest path sequence traversing links $\{1, 4\}$. For every link labeled $i \in \mathcal{L}$ that appears in the shortest path sequence, we assign $v_i^0 = v_i$, and for all other links $i \in \mathcal{L}$ not in the sequence, we set $v_i^0 = \alpha_i$. Then we obtain $v^0 = (2, 1, 1, 2, 1)$ and let $F_1^* = F_1 = [v_0, v] = [(2, 1, 1, 2, 1), (2, 2, 2, 2, 2)]$, or more compactly $F_1 = \Omega[2, :, :, 2, :]$ using our notation. Note that F_1 is a feasible hyper-rectangular subset of Ω , while I_1^* from Equation 6 is the empty set in this iteration. We now obtain 2 disjoint nonempty subsets $U_{(0,j)}$ from Equation 8, namely, $U_{(0,1)} = \Omega[1, :, :, :, :]$ and $U_{(0,4)} = \Omega[2, :, :, 1, :]$. At the end of this iteration, we move F_1 to list of feasible sets F

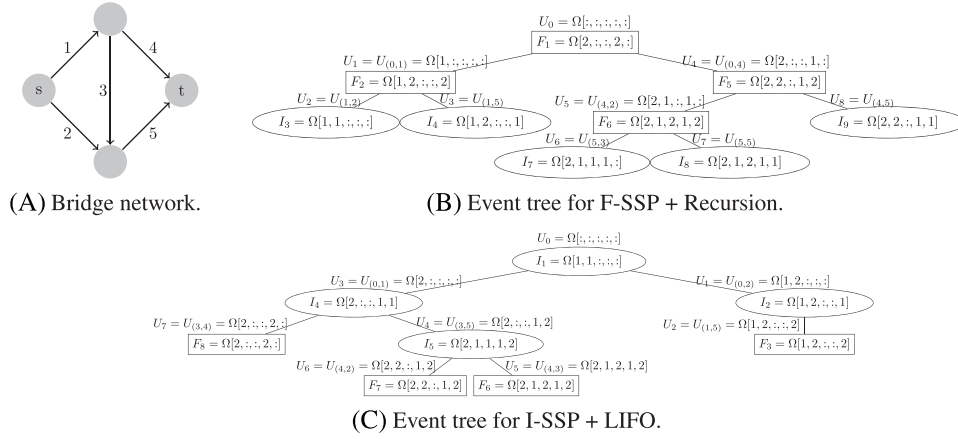


FIGURE 2 (A) Example network and decomposition trees obtained via (B) feasible-based state-space partition (F-SSP) and (C) infeasible-based state-space partition (I-SSP). Rectangles indicate feasible subsets in F , and ellipses indicate infeasible subsets in I

and *push* $U_{(0,1)}$ and $U_{(0,4)}$ to the back of list of unexplored subsets U in *reversed* order; ie, $U = [U_{(0,4)}, U_{(0,1)}]$. We continue decomposing Ω with iteration 2 using the *last* input subset from U , this is $U_1 = U_{(0,1)}$. We set $v = \beta(U_1) = (1, 2, 2, 2, 2)$ and find a deeper state v^0 . Note that link 1 is failed in subproblem U_1 . Using subroutine *feasible_v0*(v, α, β) as showed earlier, we find $v^0 = (1, 2, 1, 1, 2)$. Thus, $F_2^* = F_2 = [v^0, v]$, I_2^* is empty, and U_2^* is derived again from Equation 8, which yields subsets $U_{(1,2)} = \Omega[1, 1, :, :, :]$ and $U_{(1,5)} = \Omega[1, 2, :, :, 1]$. We move F_2 to F and *push* $U_{(1,2)}$ and $U_{(1,5)}$ to the back of U in *reversed* order; this is $U = [U_{(0,4)}, U_{(1,5)}, U_{(1,2)}]$. If we continue with iteration 2 using subproblem $U_2 = U_{(1,2)}$, we find that $\beta(U_2)$ is infeasible; thus, we can set $I_3^* = I_3 = U_2$ while sets F_3^* and U_3^* are empty. When we carry-on this process until U is empty, we fully decompose the state-space Ω into disjoint feasible subsets $F_i \in F$ and infeasible subsets $I_i \in I$. The full decomposition can be visualized with an event tree (Figure 2B), where tree nodes and leaf nodes represent feasible subsets and infeasible subsets, respectively.²⁶ Note that the tree (Figure 2B) shows a subset U_{i-1} above derived subsets F_i or I_i to illustrate their origin subproblem.

In the previous example, the removal of *last* subset in U for new subproblem decomposition and *reversed-order-pushing* of subsets $U_{(i,j)}$ in U was used to model a recursive decomposition approach (Figure 2B). Thus, an F-SSP algorithm with Recursion handling of U results into the method proposed by Li et al.¹⁷ Moreover, an implementation of an I-SSP algorithm with Recursion handling of U results into the method introduced by Liu et al.¹⁸ However, adopting a Recursion strategy is almost never advised. Alternatives, such as the LIFO strategy without *reversed-order-pushing* results into smaller $|U|$ when space complexity is a concern, and the priority heap results into faster bounds convergence when narrowing P_U is the main concern, as confirmed in our numerical experiments and the literature.²⁰

Two-terminal feasible-flow network reliability: Perhaps the earliest flow reliability related work was by Frank et al²⁷ in the context of telecommunication networks; however, the case of 2-terminal maximum flow with link capacities as discrete random variables was formalized by Evans²⁸ using a graph theory approach. Most methods for solving this problem rely on a priori enumeration of minimal paths and minimum cuts.²⁹ Nevertheless, the SSP method introduced by Doulliez and Jamouille,¹³ corrected by Alexopoulos¹⁴ and extended by Jacobson,¹⁵ does not require precomputation and exhibits superior performance in various applications.

To adapt this problem to formulae in Section 3.1, we assume every link $i \in \mathcal{L}$ has capacity levels $u_i \in \{x_i(1) < \dots < x_i(l_i)\}$ with respective probabilities $\{p_i(1), \dots, p_i(l_i)\}$. Here, the utility function $u_k(v)$ measures the maximum flow from s to t for $s, t \in V_k$, and there is a performance threshold D_k such that for $u_k(v) < D_k$, we say $v \in \Omega$ is infeasible, and feasible otherwise.

State-space partition algorithms compute maximum s - t flows and minimum s - t cuts. Again, as example application, we consider the bridge network in Figure 2A, link capacity levels $u_i \in \{0 < 1\}$ with respective probabilities $\{p_e, 1 - p_e\}$, and performance threshold $D_k = 1$. This time, we arbitrarily choose an I-SSP algorithm with LIFO handling of U . We begin with $F = [], I = []$ and $U = [\Omega]$. The first iteration starts extracting the *last* subset from U and setting $[\alpha, \beta] = U_0 = \Omega$. Efficient maximum flow and minimum cut algorithms can be found elsewhere.³⁰ After verifying that α is infeasible (ie, the maximum flow from s to t is less than 1), we set $v = \alpha$. An implementation of *infeasible_v0*(v, α, β) for deriving a deeper state v^0 is as follows. Compute the minimum s - t cut $C_k(v) \subseteq \mathcal{L}$, say $C_k(v) = \{1, 2\}$, and raise the capacity levels of the components outside the cut to their maximum capacity levels ($v_i^0 = \beta_i, \forall i \notin C_k(v)$) while the capacities of components

in $C_k(v)$ are raised from $v_i^0 = v_i$ towards β_i such that the sum of capacities $u_i \forall i \in C_k(v)$ is always below the performance threshold D_k . Since raising the capacity level of any component in $C_k(v)$ would result into feasibility, for this iteration, we directly find $v_i^0 = v_i, \forall i \in C_k(v)$. Then we set $I_1 = [v, v^0] = \Omega[1, 1, :, :, :]$ and move it to I and derive nonempty unexplored subsets $U_{(1,1)} = \Omega[2, :, :, :, :]$ and $U_{(1,2)} = \Omega[1, 2, :, :, :]$ (using Equation 9) and *push* them at the end of U in that order; thus, $U = [U_{(0,1)}, U_{(0,2)}]$. We proceed extracting the *last* subset from U and set $[\alpha, \beta] = U_1 = U_{(0,2)}$. We verify that α is infeasible and set $v = \alpha$. Using the subroutine **infeasible_v0**(v, α, β), we derive $v^0 = (1, 2, 2, 2, 1)$. Thus, $I_2 = \Omega[1, 2, :, :, 1]$, and Equation 9 yields a single nonempty subset $U_{(1,5)} = \Omega[2, 1, :, :, 2]$ that we *push* to the end of U . The next iteration takes the last subproblem $[\alpha, \beta] = U_2 = U_{(1,5)}$, where α is feasible, and thus, we set $F_2 = [\alpha, \beta]$. When carrying on this process until U is empty, we obtain all disjoint subsets $I_i \in I$ and $F_i \in F$ represented in the event tree of Figure 2C. In contrast to Figure 2B, I-SSP algorithms derive infeasible subsets at tree nodes and feasible sets at leaf nodes.

Note that in the previous example, we used a LIFO handling of U , resulting into a different numbering order of subsets with respect to Figure 2B. Also, note that the previous example represents a reduction to the s - t network reliability due to unit-demand and unit-capacity links. Besides simplicity, this is to stress that literature of multistate systems has been available for some time, and overlooking such advancements leads to independent developments of reliability methods for binary systems with substantial overlapping (eg, I-SSP using Recursion¹⁸). Also, with *coherence* as an assumption, Barlow et al²² generalized the theory of binary systems to multistate systems in which the system can have different performance levels.

Besides network reliability problems mentioned in this subsection, Jacobson¹⁵ studied stochastic spanning trees in a setup that includes the well-known all-terminal reliability problem as a specific case. Also, Daly²¹ studied stochastic multiterminal and multicommodity network flow problems.

3.4 | RAILS: a new framework for interdependent network reliability problems

In this subsection, we extend the single system *network reliability* problem to the SOS. We term the new framework RAILS. Consider a set of infrastructures K represented as stochastic graphs $G_k(V_k, E_k), \forall k \in K$ and the ensemble $G(V, E)$ with set of nodes $V = \cup_{k \in K} V_k$ and set of links $E = \cup_{k \in K} E_k$. As before, we consider an arbitrary labeling of all components in the ensemble $\mathcal{L} = \{1, \dots, a\}$ with $a = |V| + |E|$ and use one-to-one mappings $\psi : V \mapsto \mathcal{L}$ and $\phi : E \mapsto \mathcal{L}$ whenever we require specifying node or link labels, respectively. The state-space Ω in the SOS setting is equally defined in Equation 2. In Section 3.1, we relied on system-specific utility function $u_k(\cdot)$ and performance threshold D_k to define the feasible and infeasible domains as well as computing network reliability r (Equation 3). Let us now introduce the widely used structure function $SF_k : \Omega \mapsto \{0, 1\}$ such that $SF_k(v)$ outputs 1 if $u_k(v) \geq D_k$ and 0 otherwise. A very simple extension of the structure function to the SOS case is $SF(v) = \prod_{k \in K} SF_k(v)$ with $v \in \Omega$. Despite the simplicity of $SF(\cdot)$, it requires input from various stakeholders to establish acceptable performance level thresholds D_k for all infrastructure services $k \in K$ during contingencies. The *systemic reliability*, denoted R , is the probability that all performance thresholds $D_k, \forall k \in K$ can be met. We can estimate R as follows:

$$R = \Pr[SF(v) = 1] = \sum_{v \in \Omega} \Pr[v] I_{\mathcal{F}}(v) = 1 - \sum_{v \in \Omega} \Pr[v] I_{\mathcal{I}}(v), \quad (10)$$

where \mathcal{F} is the feasible domain $\mathcal{F} = \{v \in \Omega : SF(v) = 1\}$, \mathcal{I} is the infeasible domain $\mathcal{I} = \{v \in \Omega : SF(v) = 0\}$, and $I_S(\cdot)$ is the indicator function that outputs 1 if v is in the domain S and 0 otherwise. Estimating R from Equation 10 poses the same computational challenges described in Section 3.1 for Equation 3. Thus, we can use efficient methods such as the SSP method for bounding R with similar inequalities to Equation 7.

In this paper, we consider the problem of feasible multicommodity flows with interdependency requirements and use a mixed-integer linear program to model it. The motivation for using mathematical programming is twofold. First, many practical applications in reliability and systems engineering can be stated as a network flow-based formulation where mathematical programming can serve as a common language across disciplines when considering interdependent systems.³¹ Also, network design of interdependent networks as well as their joint restoration has been modeled using a similar structure.³²⁻³⁴ Second, optimization models guarantee optimality when solutions are found and the development of general purpose optimization solvers remains an active area of research.³⁵ We consider multiple source and sink nodes; however, after using a reduction, the problem we consider can be seen as a stack of $|K|$ 2-terminal feasible-flow problems (one for every system $k \in K$). Note that some components in network $k \in K$ may require services provided by another network $\bar{k} \in K$ such that $k \neq \bar{k}$, thus coupling otherwise independent network-flow problems. As before, we consider the ensemble $G(V, E)$. Also, a component labeled $i \in \mathcal{L}$ has capacity u_i that is a discrete random variable taking values from

the set $\{x_i(1) < \dots < x_i(l_i)\}$ with respective probabilities $\{p_i(1), \dots, p_i(l_i)\}$. Moreover, each infrastructure $k \in K$ has an associated set of commodities L_k , and each node $i \in V$ has a demand b_{il_k} . When $b_{il_k} > 0$, we say that node $i \in V$ is a customer of commodity $l_k \in L_k$. Conversely, when $b_{il_k} < 0$, we say that node $i \in V$ is a supplier of commodity $l_k \in L_k$ and a transshipment node otherwise. Certain nodes $i \in V$ will require satisfying demands $b'_{il_k} > 0$ to function properly, and their capacity level $u_{\psi(i)}$ will be factored by respective interdependent functionality $w_i \in \{0, 1\}$. In addition to this, flow f_{ijl_k} of commodity $l_k \in L_k$ can traverse links $(i, j) \in E_k, k \in K$. The *feasible* performance of the SOS is verified when $u_k(v) \geq D_k$ for all networks $k \in K$, where $u_k(v)$ is the sum of all commodity units in L_k reaching customer nodes. We will compute $u_k(v)$ using optimization model in Equations 11a-g however, we first consider 3 reductions. First, we introduce super source node s_k , super sink node t_k , and a directed link (t_k, s_k) with fixed unbounded capacity for all networks $k \in K$. Note that a fictitious node or link is added to respective sets V_k or E_k , labeled in \mathcal{L} , and assumed perfectly reliable. Second, for every customer node (with $b_{il_k} > 0$), we introduce a directed link from such node to t_k and assume capacity b_{il_k} . We do the same for every supplier node (with $b_{il_k} < 0$), but directed links go from s_k to such nodes and assume nonnegative capacity $-b_{il_k}$. Third, we add directed links from nodes with required demands b'_{il_k} to sink t_k and assume capacity b'_{il_k} . We consider that every link in E has an associated cost $c_{ij} = 0$ except for edges (t_k, s_k) , which have cost -1 .

$$\text{Min } \sum_{k \in K} \sum_{(i,j) \in E_k} \sum_{l_k \in L_k} c_{ij} f_{ijl_k} \quad (11a)$$

$$\text{s.t. } \sum_{j:(i,j) \in E_k} f_{ijl_k} - \sum_{j:(j,i) \in E_k} f_{jil_k} = 0, \quad \forall i \in V, \forall l_k \in L_k, \forall k \in K \quad (11b)$$

$$\sum_{l_k \in L_k} f_{ijl_k} \leq u_{\phi(i,j)} w_i \quad \forall (i, j) \in E_k, \forall k \in K \quad (11c)$$

$$\sum_{l_k \in L_k} f_{ijl_k} \leq u_{\phi(i,j)} w_j \quad \forall (i, j) \in E_k, \forall k \in K \quad (11d)$$

$$\sum_{k \in K} \sum_{j:(i,j) \in E_k} \sum_{l_k \in L_k} f_{ijl_k} \leq u_{\psi(i)} w_i \quad \forall i \in V \quad (11e)$$

$$w_i b'_{il_k} \leq f_{it_k l_k} \quad \forall l_k \in L_k, \forall k \in K, \forall i \in V \quad (11f)$$

$$f_{ijl_k} \geq 0 \quad \forall l_k \in L_k, \forall (i, j) \in E_k, \forall k \in K. \quad (11g)$$

Negative costs $c_{t_k s_k} = -1$ and null costs for all other flows in the objective function of Equation 11a maximize the flow of commodities from supplier nodes to demand nodes, which in turn maximizes the utility of each network. Simplifying vanishing terms in Equation 11a and grouping same network k terms, we recover the utility function $u_k(v) = \sum_{l_k \in L_k} f_{t_k s_k l_k}$. The set of constraints in Equation 11b ensures that node inflow and outflow of commodities is in equilibrium. The set of constraints in Equation 11c-d ensures that the amount of commodities traversing links does not exceed their capacities while taking into account the functionality of end nodes. The set of constraints in Equation 11e takes into account node capacity. The set of constraints Equation 11f ensures that nodes with nonsatisfied interdemands have null functionality. Constraints in Equation 11g enforce nonnegativity of flows.

The RAILS problem with utility function computed from Equation 11a can be solved using Algorithms 1 and 2. Unfortunately, subroutines *feasible_v0*(\cdot) and *infeasible_v0*(\cdot) for the 2-terminal feasible-flow problem cannot be used when considering functional interdependency constraints. Next, we extend the SSP method to obtain maximum flows and minimum cuts using Equations 11a-g and introduce problem-specific subroutines for deriving deeper state v^0 for Algorithms 1 and 2, but now for interdependent networks.

Maximum flows/minimum cuts for interdependent flow networks: Let us define u'_i to be the *effective capacity* of component $i \in \mathcal{L}$, which is computed as the nominal capacity u_i times its performance w_i . In this study, a link's effective capacity is simply $u'_i = u_i$; however, a node's effective capacity is $u'_i = u_i \cdot w_{\psi^{-1}(i)}$ due to functionality impacted by nonsatisfied interdemands. To understand how interdependent constraints complicate the issue of determining the maximum flow/minimum cut, note first that for the single network k case and assuming $w_i = 1, \forall i \in V$, Equations 11a-d and g maximize the flow of commodities from s_k to t_k .³¹ Also, using flows $f_{ijl_k}, \forall (i, j) \in E_k, l_k \in L_k$, one can derive a minimum cut from constructing a residual graph where each link capacity is $u_{\phi(i,j)} - \sum_{l_k \in L_k} f_{ijl_k}$ and in which null residual capacity links are removed. Then a breadth-first-search or depth-first-search algorithm is used to traverse the graph $G_k(V_k, E_k)$ from node s_k to derive a set of visited nodes $S_k \subseteq V_k$ and nonvisited nodes $T_k = V_k \setminus S_k$.³⁰ Also, node capacity and functionality can be handled using the graph transformation depicted in Figure 3, and inflow or outflow can

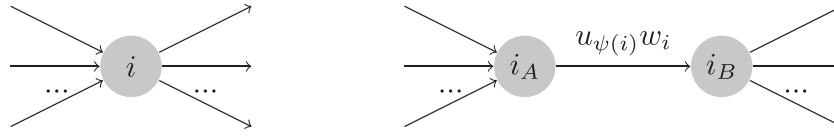


FIGURE 3 (Left) Generic node $i \in V$ with arbitrary number of in- and out-links. (Right) Graph transformation on node i to consider capacity $u_{\psi(i)} w_i$

be subtracted from the fictitious link's capacity to obtain residual capacity. Let us call $C_k(v) \in \mathcal{L}$ the minimum cut set, ie, the components with end nodes belonging to subsets S_k and T_k . We will refer to the previous process for deriving an s - t minimum cut in graph G_k as **mincut**(G_k, s_k, t_k, f, u, w). Also, let us define $W_k(v)$ to be the sum of component capacity levels u_i for all components in $C_k(v)$. Similarly, let us define $iW_k(v)$ to be the sum of component effective capacities u'_i for all components in $C_k(v)$. For a single-independent network k , it holds that $u_k(v) = iW_k(v) = W_k(v)$ from the maximum flow/minimum cut theorem. However, when many networks are considered and some interdependent demands b'_{i_k} are nonzero, then it holds that $u_k(v) = iW_k(v) \leq W_k(v)$. In other words, interdependent demands that cannot be met can induce a smaller maximum flow (minimum cut) by reducing a node's effective capacity with respect to its nominal one. In the SOS setting, we let $C(v)$ be the ensemble of minimum cut sets $C_k(v)$ for all networks $k \in K$. Whenever $iW_k(v) < W_k(v)$, we call $C_k(v)$ an *interdependency induced cut* (IIC), and we call any component with lesser capacity than its nominal due to unsatisfied demands a *cascade failing component* (CFC).

In the classic case, a minimum s - t cut set can be seen as a subset of links limiting the maximum flow between nodes s and t . In addition to $C_k(v)$, in the case of an IIC, we will need to compute a complementary cut, denoted $C'_k(v)$, which is a set of components limiting the capacity of CFCs in $C_k(v)$. Due to dynamics of interdependent networks, $C'_k(v)$ may contain components belonging to network k and/or other networks in K . Algorithm 3 shows how to determine complementary cuts $C'_k(v) \forall k \in K$, where $C(v)$ contains $|K|$ minimum cut sets $C_k(v)$.

Algorithm 3 Finding complementary cuts

```

1: procedure I-MINCUT( $G, f, u, w, C(v), CFC$ )
2:   Initialize:  $propagate = True, \forall i \in V$  let  $w'_i \leftarrow 1$ , empty sets  $C'_k(v)$  for all  $k \in K$ , empty set  $CFC(v)$ , and map data structure  $c : CFC(v) \mapsto P(\mathcal{L})$ , where  $P(S)$  is the power set of  $S$ . Apply graph transformation in 3 for unreliable and
   independent nodes.
   Comment: Find unsatisfiable nodes, enforce functionality loss  $w'_i = 0$ , and propagate failures.
3:   while  $propagate$  do
4:      $propagate \leftarrow False$ 
5:     for  $i \in CFC : w'_i = 1$  do
6:       for  $k \in K : \exists l_k \in L_k(b'_{l_k} > 0)$  do
7:          $c(i) \leftarrow \text{mincut}(G_k, S_k, T_k, f, u, w')$ 
8:         if  $\text{size}(c(i)) < b'_{i_k}$  then
9:           for  $j \in c(i) : j \in CFC$  do  $c(i) \leftarrow (c(i) \setminus j) \cup c(j)$ 
10:        end for
11:         $propagate \leftarrow True$ , and let  $w'_i = 0$ 
12:        break and goto step 5.
13:      end if
14:    end for
15:  end while
16:  Comment: While there are CFC nodes in the minimum cut  $C_k(v)$  add pertinent complementary cuts
17:  for  $C_k(v) \in C(v)$  do
18:    while  $(C_k(v) \cap CFC) \neq \emptyset$  do
19:      for  $i \in C_k(v) : i \in CFC$  do
20:         $C'_k(v) \leftarrow (C'_k(v) \setminus i) \cup c(i)$ 
21:      end for
22:    end while
23:  end for
24: end procedure
25: end procedure

```

\triangleright where $i' = \psi^{-1}(i)$
 $\triangleright k$ such that and interdemand exists
 \triangleright If unsatisfiable

Deriving deeper feasible and infeasible states v^0 : F-SSP algorithms derive a deeper feasible state v^0 by decreasing capacity levels of components not in $C(v)$ while leaving enough capacity to allow flows obtained from optimization (ie, Equations 11a-g) in addition, they decrease capacity levels of components in $C(v)$ while ensuring that the sum of new capacities of components in $C(v)$ remains at D_k or above for all $k \in K$ such that the resulting v^0 is also feasible. However, for interdependent networks, reducing the capacity level of one component in $C_k(v)$ can cause an IIC in another network \bar{k} causing it to be infeasible or induce a cascade of IICs across the SOS that could result into $iW_k(v^0) < D_k$ for some $k \in K$. Thus, when reducing capacity levels within $C_k(v)$ and $C'_k(v)$ for all $k \in K$, we need to rerun the optimization algorithm for each component capacity decrement one-by-one to ensure that v^0 remains feasible; otherwise, we step back and move on with the next component.

The derivation of a deeper infeasible state for a single network k serves us to understand the base case and increased complexity of interdependent systems. In the single network case, this consists of finding a minimum cut set $C_k(v)$ at state v . Moreover, components' capacity levels outside the cut ($i \notin C_k(v)$) are increased to β_i , whereas capacity levels of components in $C_k(v)$ are increased with the sum of their capacities remaining below performance threshold D_k . This ensures that the resulting v^0 is also infeasible. However, in the interdependent case, keeping the sum of nominal or effective capacity levels of elements in $C_k(v)$ below D_k is not enough, since one or more IICs can be due to one or more components in another network's minimum cut $C_{\bar{k}}(v)$. Thus, after obtaining $C'_k(v), \forall k \in K$, we increase the capacity levels of components outside $C(v)$ and outside $C'(v)$ all the way towards β and increase the capacity levels of components in $C(v)$ and $C'(v)$ one-by-one testing for feasibility of the resulting v^0 such that if it becomes feasible, we step back and move on to the next component.

Likelihood-based (LB) deeper state derivation: In the context of s - t network reliability and F-SSP algorithms, the work by Lim et al²⁴ introduced the *most reliable path* to derive a deeper point v^0 that results into feasible subset $[v^0, \beta]$ that has a larger contribution to shrinking bounds in Equation 7 with respect to classic shortest path.¹⁷ Let $path \subseteq \mathcal{L}$ be a set of components traversed in a path sequence from node s to t . Also, conditioned on state v in subset $U_{i-1} = [\alpha, \beta]$, component reliabilities are $r_i = p_i(v_i) / \sum_{j=\alpha_i}^{\beta_i} p_i(j)$. Moreover, the likelihood $L(path)$ of a generic path is simply the product of component reliabilities in $path$, ie, $L(path) = \prod_{i \in path} r_i$. Lim et al²⁴ proposed a modified Dijkstra algorithm that computes the path with maximum likelihood. However, since $L(path)$ is a nonlinear function of component probabilities, we can use the negative log-likelihood (NLL) function and obtain a simpler expression $NLL(path) = \sum_{i \in path} -\log(r_i)$. Note that minimizing $NLL(\cdot)$ is equivalent to maximizing $L(\cdot)$. Moreover, a $path$ obtained using classical Dijkstra's algorithm with link weights $c_{ij} = -\log(r_{ij}) - \log(r_j)$ minimizes $NLL(v)$ while accounting for both link and node reliabilities r_{ij} and r_j , respectively. We extend this strategy to our RAILS problem by considering costs c_{ij} as described in Equation 11a. However, to ensure that utilities $u_k(v) = \sum_{l_k \in L_k} f_{t_k, s_k, l_k}$ are maximized, we consider that fictitious link (t_k, s_k) has weight $c_{t_k, s_k} = -(1 + c_{max}) \cdot a$, where c_{max} is the maximum link cost c_{ij} as defined above among all links in $(i, j) \in E$ and $a = |\mathcal{L}|$.

The flows f obtained from Equations 11a-g are used directly by subroutine **feasible_v0**(\cdot) to derive most reliable paths or flows. However, for its counterpart **infeasible_v0**(\cdot), we will need to do more work for deriving minimum cuts with maximum likelihood. In particular, we exploit the fact that the minimum cut is a subset of *saturated links* when computing maximum s - t flow, and we obtain a maximum likelihood minimum cut with a 2-level minimum s - t cut algorithm. First, the maximum s - t flow is computed to obtain flows f (eg, using Equations 11a-g), and a residual graph is constructed where each link has capacity $u_{\phi(i,j)} - \sum_{l_k \in L_k} f_{ij|l_k}$. Then we consider a secondary maximum s - t flow problem where every saturated link (residual capacity zero) has capacity $c_{ij} = u_{\phi(i,j)}$ and nonsaturated links have unbounded capacity and compute the minimum cut $C_k(v)$ as described before with routine **mincut**(G_k, s_k, t_k, f, u, w). Finally, $C(v)$ is the minimum cut with maximum likelihood. We will adopt these LB strategies in our numerical experiments and show how they affect convergence of bounds in Equation 7.

3.5 | RAILS modeling capabilities and connection with risk

The multicommodity network flow problem with interdependency constraints in Equations 11a-g can be used for modeling a variety of practical problems. Before we describe such applications, we recall that the ensemble of systems $G(V, E)$ has set of nodes $V = \bigcup_{k \in K} V_k$ and set of links $E = \bigcup_{k \in K} E_k$, and each network system has associated set of commodities L_k . Note that these sets do not need to be disjoint for extra lifeline modelers' advantage.

Just as fictitious network components are added to model more complicated problems (eg, super source node for multiple generation nodes), we can consider fictitious networks to model *backup systems* supporting the operation of networks. In particular, a network $k \in K$ with interdependent demands $b_{i|k}$ such that $l_{\bar{k}} \in L_{\bar{k}}, i \in (V_k \cap V_{\bar{k}})$, and $k, \bar{k} \in K$, can be provided with a backup system modeled by a network $G_{\bar{k}}(V_{\bar{k}}, E_{\bar{k}})$ with null performance threshold ($D_{\bar{k}} = 0$) since its only function is to support other components in $k \in K$ such that $\bar{k} \neq k$. $V_{\bar{k}}$ will contain as many supply nodes as backup generation facilities and $E_{\bar{k}}$ will contain directed links from backup generation to consumers as provided in the backup design. Note that even though commodities $L_{\bar{k}}$ are associated to infrastructure \bar{k} , there is no restriction in our formulation preventing $L_{\bar{k}} \cap L_{\bar{k}} \neq \emptyset$, eg, a backup power network has commodities that also appear in the main power network and both can power the same node. This simple yet powerful abstraction allows for modeling unreliable backups, their shared use among components, and optimal allocation of resources.

Cyber and *logical* interdependencies can also be modeled using the fictitious network approach. For instance, a set of nodes $p \in V$ may require that other nodes in $q \in V$ are functional for them to be functional; this is easily handled by adding a fictitious network $G_{\bar{k}}(V_{\bar{k}}, E_{\bar{k}})$ with $V_{\bar{k}} = p \cup q$, fictitious set of commodities $L_{\bar{k}} = \{l_{\bar{k}}\}$, and where nodes in q are

suppliers and nodes in p are customers. Fictitious-directed links are added to E_k connecting each supplier to each customer and demand intensities, while link capacities can be selected homogeneously or heterogeneously to model AND, OR, and XOR and more general interdependencies.

Reliability estimation can be seen as a specific case of *risk estimation* where a feasibility threshold is established (ie, D_k), and there are associated losses for being below or not a performance level. Nevertheless, one could extend the reliability assessment to a risk assessment context by considering multiple thresholds D_k separately and use the combined by-products to build performance annual exceedance probability curves that can be easily linked to associated losses. We take this approach for risk estimation in our numerical experiments.

While the core of this contribution is to provide an alternative methodology to dealing with multiple FMPs by exploiting network structure, as opposed to reducing the set of FMPs as in past research, we also want to highlight the applicability of RAILS as a stand-alone analytical framework for studying networked engineering systems under multiple hazards providing sound theoretical bounds on reliability and risk estimates. In addition to this, RAILS by-products can be used to compare retrofit strategies and backup provisions, as well as to rank components via reliability-based importance metrics.¹⁹

4 | COMPUTATIONAL EXPERIMENTS

We begin this section by showing how different SSP strategies impact convergence of bounds in Equation 7 as well as the size of U in the simpler setting of a single infrastructure and s - t network reliability. Then we test new SSP algorithms for the RAILS problem considered in this paper, namely, 2-terminal feasible flows with interdependency constraints. We round up this section with an application of RAILS to seismic performance loss, using realistic systems artificially positioned in the San Francisco Bay Region (SFBR).

We implement SSP Algorithms 1 and 2 in a Python prototype and use Gurobi³⁶ for solving Equation 11a–g.

4.1 | Comparison of SSP algorithms and impact on convergence of bounds and $|U|$

We first consider the network in Figure 4 (left) with unreliable nodes that can fail with probability 10%. \mathcal{L} is consistent with the labeling shown. The exact s - t network reliability is 0.787052 and can be computed using any SSP algorithm in Table 1. Figure 4 (center) shows how bounds converge as F-SSP algorithms run considering different policies for handling U . We see that Heap handling of U prevails over other strategies when aiming at tight bounds. Also, using LB further tightens bounds as depicted by Heap with and without considering LB. Figure 4 (right) shows a comparison among A1 and A2 from Table 1. As expected, the upper bound converges faster when using A1 and the lower bound converges faster when using A2. Algorithm A6 in the same figure does not feature bounds that converge nearly as fast as for A1 or A2; however, it does perform better than its F-SSP counterpart A5 in Figure 4 (center).

Figure 5 compares algorithms in Table 1 as measured by their precision and memory consumption, ie, P_U at termination and maximum value of $|U|$ during execution. We run each algorithm for 1,000 iterations as the bounds tighten enough by then in the experiments to unravel algorithmic differences. We also vary the size parameter of the grids (N) to show its effect on the performance of algorithms. Figure 5 (left) shows the final size as well the location of the gap P_U containing

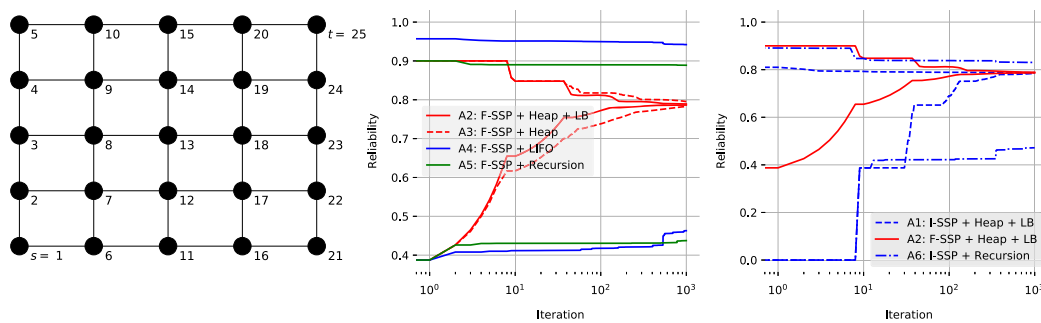


FIGURE 4 5 × 5 Grid network (left), comparison among feasible-based state-space partition (F-SSP) algorithms (center), and comparison between F-SSP and infeasible-based state-space partition (I-SSP) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 List of SSP configurations explored in Section 4.1

| Algorithm | Configuration | References | Algorithm | Configuration | References |
|-----------|-------------------|---|-----------|-------------------|--|
| A1 | I-SSP + Heap + LB | This paper | A4 | F-SSP + LIFO | This paper and Alexopoulos ²⁰ |
| A2 | F-SSP + Heap + LB | This paper and Lim and Song ²⁴ | A5 | F-SSP + Recursion | Li and He ¹⁷ |
| A3 | F-SSP + Heap | This paper and Alexopoulos ²⁰ | A6 | I-SSP + Recursion | Liu et al ¹⁸ |

Abbreviations: F-SSP, feasible-based state-space partition; I-SSP, infeasible-based state-space partition; LB, likelihood-based; LIFO, last-in-first-out; SSP, state-space partition.

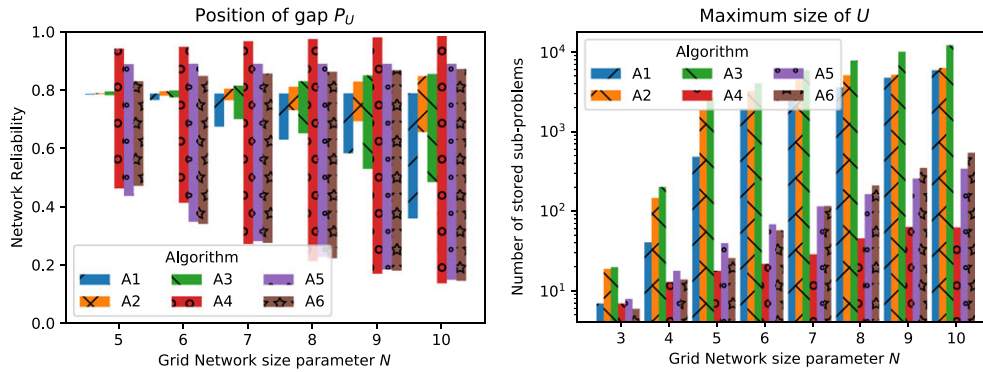


FIGURE 5 Final values of P_U (left) and maximum values of $|U|$ (right), when running various algorithms on grid networks of size $N \times N$. Each algorithm ran 1,000 iterations [Colour figure can be viewed at wileyonlinelibrary.com]

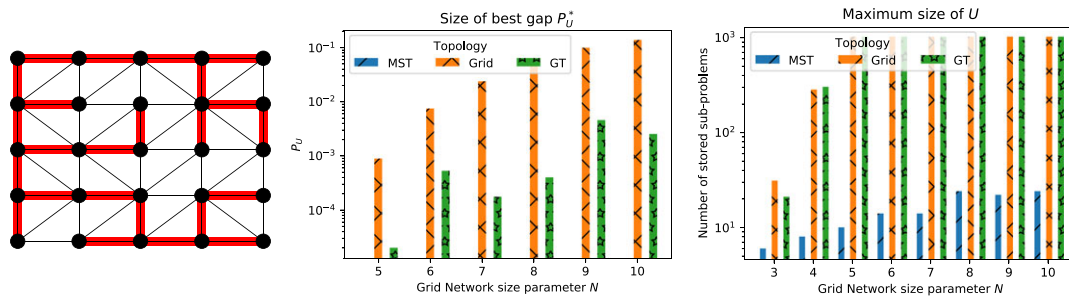


FIGURE 6 Left: 5×5 grid with extreme topologies GT (thin line) and MST (thick line). Best values of P_U (center) and maximum values of $|U|$ (right), while varying parameter N . Each algorithm ran 1,000 iterations [Colour figure can be viewed at wileyonlinelibrary.com]

network reliability estimates for all SSP algorithms and sizes $N \times N$. It is clear that algorithms with Heap handling of U achieve smaller gaps, particularly when using LB. Algorithm A2 provides the smallest gap, while the best upper and lower bounds are consistently given by A1 and A2, respectively. Figure 5 (right) turns the attention into memory consumption measured as the maximum number of stored subproblems at any time during algorithm execution. We observe that Heap algorithms have the largest memory consumption, revealing a memory/precision trade-off. Furthermore, consistent with previous findings, the LIFO strategy seems to be the most memory efficient way to handle U .

Figure 6 further explores the effect of topology. We consider the minimum spanning tree (MST) and greedy triangulation (GT) as extreme models of sparse and redundant planar networks, respectively (Figure 6 [left]). This time, we obtain the gap P_U (see Figure 6 [center]) from our best upper and best lower bound estimates, which are consistently provided by A1 and A2, respectively. The maximum value of $|U|$ is in turn the maximum among such values encountered in A1 and A2 (Figure 6 [right]). Also, we see that the MST networks are solved exactly (Figure 6 [center]).

4.2 | Interdependent grid networks

To further test the efficiency of SSP algorithms, we consider a model of synthetic interdependent networks with grid topology. Also, we will consider various performance threshold levels D_k and different scenarios of interconnectivity. We consider that links and nodes can fail. For ease of interpretation, we will adopt 2 capacity levels for components, namely,

the cases of complete damage and no damage. Since we will need capacitated and interconnected systems with node demands, we will use the following network model. First, define the number of infrastructures $k \in K$ as well as their size $N \times N$. Note that with this model, there are $|K|(N^2 + 2(N - 1)N)$ components. Then simulate node demands for each infrastructure $k \in K$ identifying them as customers, suppliers, etc. Also, for each infrastructure $k \in K$, we select at random a percentage I_{str} of nodes and assign interdemands $b'_{i,k'}$, such that $k \neq k'$ and connect them to major consumer nodes j that are selected at random from infrastructure k' by adding a directed interdependent link (j, i) to $E_{k'}$ with enough capacity to transport $b'_{i,k'}$. We selected 30% as the proportion of consumer and supplier nodes for each grid $k \in K$. Also, the intensity of all intrademands and supplies are set to 1, maximum capacity level of links is set to 1 with probability 90%, and for nodes, we adopt unbounded capacity with probability 95%. When components fail, their capacity levels are assumed to be 0. Figure 7 shows a realization of the model described above, in which we have incrementally added interdemands ($b'_{i,k'} = 0.001$) and cross-system links as I_{str} increases. Each experiment consists of running 1 of the 2 SSP algorithms (A1 and A2 in RAILS context) on realizations of the model while using the same magnitude of performance threshold D_k for each infrastructure $k \in K$ and a fixed value for I_{str} . We set as limit of computation 1,000 seconds.

As the size of the grids increase, we note that bounds do not converge to a meaningful precision; thus, we focus our attention to variance reduction when adopting an ISS approach on remaining subproblems in U .¹² Variance reduction of a target estimator is measured as variance of a reference estimator (eg, crude MCS) over the variance of a target estimator.³⁷ Figure 8 shows how many orders of magnitude more efficient in terms of variance reduction are SSP-derived estimators with respect to MCS. The upper bound on R from I-SSP tends to induce more variance reduction than F-SSP-based estimators. Figure 9 shows systemic reliability estimates and their variability as a function of D_k and I_{str} , as well as the variance ratio of F-SSP and I-SSP as reference and target estimators, respectively. Moreover, we observe that F-SSP tends to be better than I-SSP when the performance threshold is close to the maximum performance level of small systems; nonetheless, as the size of the systems increases, the variance ratio favors I-SSP overall.

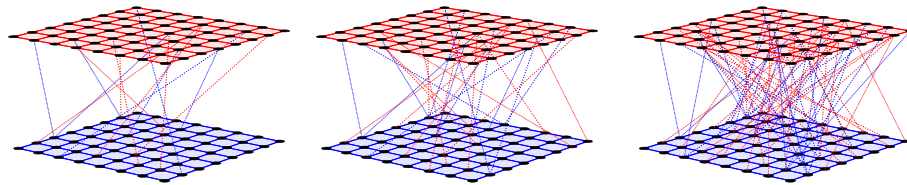


FIGURE 7 Realization of 2-layer 8×8 grids with interdependency degrees I_{str} of 10%, 20%, and 50% [Colour figure can be viewed at wileyonlinelibrary.com]

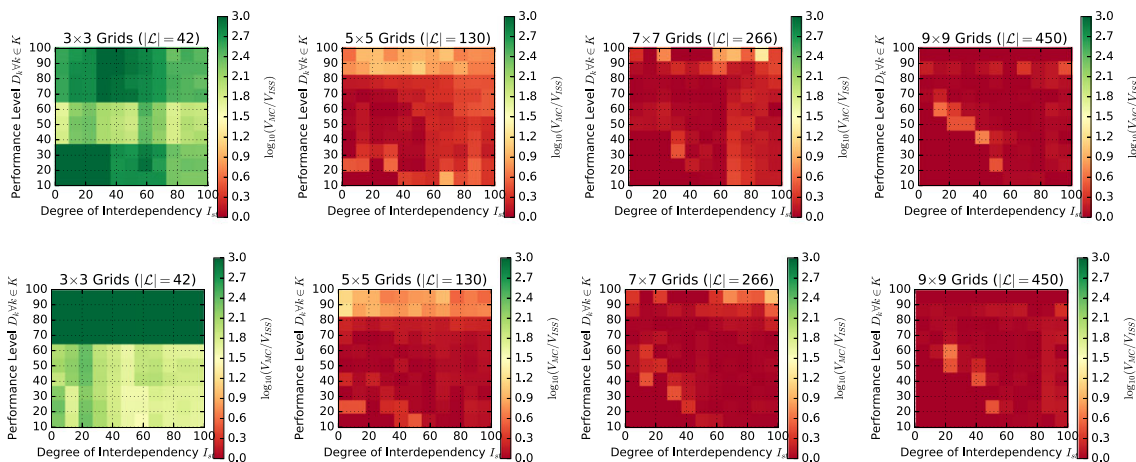


FIGURE 8 Variance ratio for crude Monte Carlo simulation estimator and importance and stratified sampling estimator. $\log_{10}(V_{MCS}/V_{ISS})$ represents how much more efficient is an importance and stratified sampling estimator with respect to Monte Carlo simulation. Top plots are obtained using infeasible-based state-space partition and bottom plots using feasible-based state-space partition [Colour figure can be viewed at wileyonlinelibrary.com]

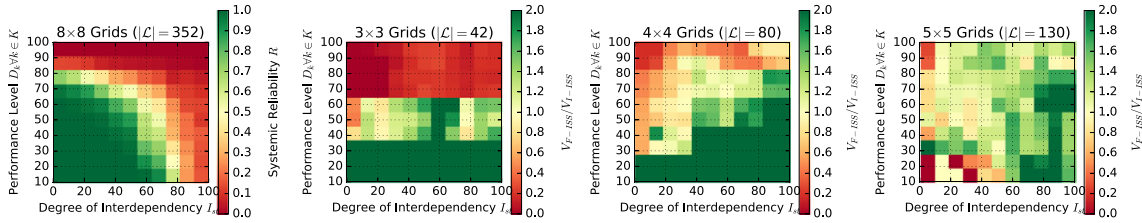


FIGURE 9 Systemic reliability R estimates for 8×8 grids using feasible-based state-space partition (F-ISSP) algorithm for different D_k and I_{str} . (Left). Other plots show the ratio of estimator variances using F-ISSP and infeasible-based state-space partition with color clipped between 0 and 2. $V_{F-ISS}/V_{I-ISS} > 1$ means that infeasible-based state-space partition induces more variance reduction than F-ISSP [Colour figure can be viewed at wileyonlinelibrary.com]

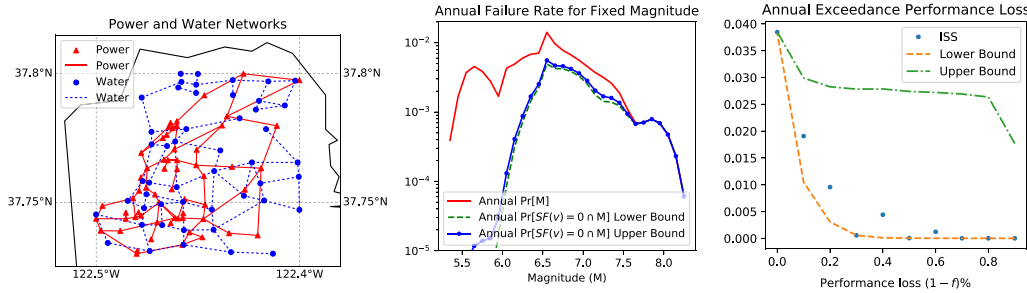


FIGURE 10 Left: topology and relative position of networks artificially positioned in the San Francisco Bay Region. Center: Annual failure rate associated to earthquake scenarios with same magnitude for $f = 1$. Right: Annual exceedance performance loss for the entire seismic catalog considering different performance loss levels $(1 - f)$. ISS, importance and stratified sampling [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Component fragility functions. $\Phi(\cdot)$ is the cumulative distribution function of a random variate with standard normal distribution

| Component Type | IDs | Fragility Function | Notes |
|---------------------|---------|---|----------------------------------|
| Power generation | 1-8, 60 | $\Phi\left(\frac{\log(\text{PGA}/\text{median})}{\beta}\right)$ | median = 0.52; $\beta = 0.55$ |
| Power distribution | 9-59 | $\Phi\left(\frac{\log(\text{PGA}/\text{median})}{\beta}\right)$ | median = 0.45; $\beta = 0.45$ |
| Water generation | 1-15 | $\Phi\left(\frac{\log(\text{PGA}/\text{median})}{\beta}\right)$ | median = 0.77; $\beta = 0.65$ |
| Water distribution | 16-49 | $\Phi\left(\frac{\log(\text{PGA}/\text{median})}{\beta}\right)$ | median = 0.68; $\beta = 0.55$ |
| Water network links | All | $2 \times 10^{-5} \cdot \text{PGV}^{2.25}$ | Units in rate of failures per km |

4.3 | Case study: risk of interdependent networks artificially positioned in the SFBR

To showcase an application of the RAILS framework for computing risk of interdependent LSs, we use a dataset of interdependent networks artificially located in the SFBR. Here, seismic activity from nearby faults can exert damage to network components, and there is uncertainty in the source of the ground motion as well as in the intensity of the quake.

The dataset of interdependent networks consists of a power subtransmission system and a potable water distribution network (Figure 10). The test systems represent simplified yet realistic networks serving Shelby County, TN, USA.³⁸ The power network has 60 nodes and 121 links, while the water network has 49 nodes and 71 links. The description of the topologies and general properties of the interdependent networks is based on the work by Duenas-Osorio et al.³⁹ Since the objective of this analysis is to showcase the application of our framework in seismic analyses, we consider component fragilities as depicted in Table 2.⁴⁰ Also, demand and capacity data as well as interdependencies were adopted from a study that uses the same dataset.⁴¹

To obtain ground motions' mean and residuals (Equation 1), we use the OpenSHA Event Set Calculator.⁴² This software outputs parameters for input locations using specified seismic source model, prediction equations, and intensity measures. We used the UCERF2 as seismic source model,⁴³ Boore and Atkinson's prediction equations,⁴⁴ and PGA and

PGV* for target parameters as required input for fragility functions. The selected source model treats faults as discretized ruptures with associated magnitudes and annual occurrence rates. We only consider seismic events with annual rates $\nu \geq 10^{-5}$ as done in a similar study.⁴ For each seismic event, we consider 100 realizations, which totals 99,400 IMPs for computing systemic performance. Considering that we do not reduce the seismic catalog, this final size is considerably larger than those in previous work (eg, 4,220 IMPs reduced to 200⁴).

To establish feasibility in this case study, we let the performance threshold D_k be the sum of all demands in infrastructure $k \in K = \{\text{Power, Water}\}$. To quantify risk, we consider a factor $f \in [0, 1]$ to be multiplied by D_k to account for various performance levels. For this case study, we conduct an experiment for f taking values from 0.1 to 1.0 with 0.1 size increments. Also, we separately consider F-SSP and I-SSP estimators. For each estimator, we compute 1,000 iterations or less within 1,000 seconds to obtain F , I , and U . Then, from U , we draw 10,000 samples ν , each with probability ω'_ν and store outcomes $SF(\nu)$. In this precomputation step, we considered failure probabilities of 5% for network links and 1% for all nodes. Finally, for each IMP_{*ij*}, associated to ES_{*i*} as defined in Section 2, we update bounds P_F and P_I to respective failure probabilities, and each precomputed sample outcome is updated as $\omega_{ij}(P_F + P_U \cdot SF(\nu)\omega'_{ij,\nu}/\omega'_\nu)$, where weight ω_{ij} is as defined in Section 2 and IS factor $\omega'_{ij,\nu}/\omega'_\nu$ corrects for the probability of occurrence of sample ν in the probability space defined by fragilities associated to IMP_{*ij*}.

Figure 10 (center) shows the annualized failure rate of the SOS conditioned on the occurrence of specific magnitude earthquakes in the catalog. $\Pr[M]$ stands for the aggregated annual rates of IMPs with same magnitude. Obviously, high-magnitude events have smaller combined annual rate. Furthermore, lower and upper bounds on $\Pr[(SF(\nu) = 0) \cap M]$ quantify the annualized SOS failure probability for earthquakes of magnitude M in the catalog. These bounds are almost indistinguishable in most regions of the plot, and they refer to performance factor $f = 1$, ie, the SOS is in a feasible state when all demands can be met. Also, note the high contribution to the SOS failure probability by ESs with magnitudes between 6.5 and 7.5, which is expected due to both their large combined annual rates of occurrence and their capacity damage at the SOS level.

Figure 10 (right) shows the annualized exceedance performance loss for the SOS when considering different performance loss levels $1 - f$ and the combined effect of the whole seismic catalog. If a loss function L is known, then the horizontal axis is simply transformed by $L(1 - f)$, and Figure 10 (right) becomes an exceedance loss curve for monetary costs. Our risk assessment provides insight on the overall annual exceedance loss of the SOS via theoretical bounds, while a more refined analysis can be performed by increasing the number of samples and number of SSP algorithm iterations, or by segmenting the seismic catalog into separate subsets of FMPs using similarity measures and developing respective estimators in parallel. The estimate based on ISS is very close to the lower bound on the failure probability, suggesting that few subsets in I will provide an upper bound close to the true value and upper bound results will tend to be rather conservative. We have noticed a similar trend when it comes to approximating unreliability (associated to lower bound) and reliability (associated to upper bound), where efficient fully polynomial randomized approximation schemes based on a few infeasible subsets exists for the former but not for the latter.⁴⁵

5 | CONCLUSIONS

In this paper, we proposed efficient analytical state-space partition (SSP) algorithms for estimating the systemic reliability of networked systems with cyclic interdependencies. Computations can be recycled and used in Monte Carlo sampling with considerable variance reduction over crude Monte Carlo methods.

We introduced a new framework for general interdependent network reliability problems for systemic performance assessment termed RAILS. Such a framework is presented using a mathematical programming problem formulation and new SSP-based algorithms for their efficient solution. Moreover, while the RAILS framework serves as a generalization of network flow-based reliability problems in the system of systems (SOS) setting, this can be used in simpler problems (eg, all-terminal and source-terminal connectivity), and the independent network case. We provided improved algorithms that integrate likelihood-based decompositions in the two main strategies proposed, feasible- and infeasible-based, and gave a systematic treatment of their implementation, while revealing their relation to existing methods.

Numerical experiments show that RAILS can be used to derive exact estimates or tight bounds for relatively small networks more efficiently (using Heap and LB strategies) than existing decomposition methods. Also, for larger systems, using partial decomposition and Monte Carlo sampling, RAILS exhibits improved performance over crude sampling. In

*Peak ground acceleration (PGA) and peak ground velocity (PGV).

addition, we exemplified the estimation of risk in realistic interdependent networks while using seismic catalogs of size considerably larger than those in the literature.

Further research will focus on improving the SSP algorithms and address the lack of knowledge on a priori guarantees of approximation as done in more recent work.⁴⁶ While we introduced problem-specific routines *feasible_v0*(·) and *infeasible_v0*(·) for network flow problems with interdependency constraints, developing routines that are completely structure agnostic would result in a direct extension of RAILS to problems beyond those in the form of Equations 11a–g. Moreover, adding physical properties, correlations and repairs would augment the applications of RAILS. Finally, an in-depth study of importance metrics that can be derived using SSP algorithms would provide new perspectives on element ranking for protection and retrofitting of interdependent systems.

ACKNOWLEDGEMENTS

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

ORCID

Roger Paredes  <http://orcid.org/0000-0003-3683-2186>

REFERENCES

- Han Y, Davidson RA. Probabilistic seismic hazard analysis for spatially distributed infrastructure. *Earthq Eng Struct Dyn*. 2012;41(11):2141-2158.
- Jayaram N, Baker JW. Efficient sampling and data reduction techniques for probabilistic seismic lifeline risk assessment. *Earthq Eng Struct Dyn*. 2010;41(11):1109-1131.
- Stergiou E, Kiremidjian AS. Treatment of uncertainties in seismic-risk analysis of transportation systems. Pacific Earthquake Engineering Research Center; 2008.
- Miller M, Baker J. Ground-motion intensity and damage map selection for probabilistic infrastructure network risk assessment using optimization. *Earthq Eng Struct Dyn*. 2015;44(7):1139-1156.
- Manzour H, Davidson RA, Horspool N, Nozick LK. Seismic hazard loss analysis for spatially distributed infrastructure in Christchurch New Zealand. *Earthq Spectra*. 2016;32(2):697-712.
- Kim Y, Kang WH, Song J. Assessment of seismic risk and importance measures of interdependent networks using a non simulation-based method. *J Earthq Eng*. 2012;16(6):777-794.
- Bruneau M, Chang SE, Eguchi RT, et al. A framework to quantitatively assess and enhance the seismic resilience of communities. *Earthq Spectra*. 2003;19(4):733-752.
- Ouyang M, Dueñas-Osorio L. Time-dependent resilience assessment improvement of urban infrastructure systems. *Chaos (Woodbury, N.Y.)* 2012;22(3):033-122.
- Liu W, Li J. An improved recursive decomposition algorithm for reliability evaluation of lifeline networks. *Earthq Eng Eng Vib*. 2009;8(3):409-419.
- Dueñas-Osorio L, Rojo J. Reliability assessment of lifeline systems with radial topology. *Comput Aided Civ Inf Eng*. 2011;26(2):111-128.
- Stern R, Song J, Work D. Accelerated Monte Carlo system reliability analysis through machine-learning-based surrogate models of network connectivity. *Reliab Eng Syst Saf*. 2017;164(2016):1-9.
- Daly MS, Alexopoulos C. State-space partition techniques for multiterminal flows in stochastic networks. *Networks*. 2006;48(2):90-111.
- Doulliez P, Jamouille E. Transportation networks with random arc capacities. *Revue française d'automatique, d'informatique et de recherche opérationnelle (Rairo)*; 1972.
- Alexopoulos C. A note on state-space decomposition methods for analyzing stochastic flow networks. 1995;44(2):354-357.
- Jacobson JA. State space partitioning methods for solving a class of stochastic network problems. *PhD Thesis*. Atlanta, GA: Georgia Institute of Technology; 1993. <http://hdl.handle.net/1853/25637>
- Dotson W, Gobien J. A new analysis technique for probabilistic graphs. *IEEE Trans Circuits Syst*. 1979;26(10):855-865.
- Li J, He J. A recursive decomposition algorithm for network seismic reliability evaluation. *Earthq Eng Struct Dyn*. 2002;31(8):1525-1539.
- Liu W, Qian Y, Li J. Minimal cut-based recursive decomposition algorithm for seismic reliability evaluation of lifeline networks. *J Earthq Eng Eng Vib*. 2007;6(1):21-28.
- Ghosn M, Dueñas-Osorio L, Frangopol D, et al. Performance indicators for structural systems and infrastructure networks. *J Struct Eng*. 2016;142(Technical Papers):1-18.
- Alexopoulos C. State space partitioning methods for stochastic shortest path problems. *Networks*. 1997;30(1):9-21.
- Daly MS. State space partition techniques for multiterminal and multicommodity flows in stochastic networks. *PhD Thesis*. Atlanta, GA: Georgia Institute of Technology; 2001. <http://hdl.handle.net/1853/25637>
- Barlow RE, Wu AS. Coherent systems with multi-state components. *Math Oper Res*. 1978;3(4):275-281.

23. Bai G. *Efficient Evaluation of Multistate Network Reliability*. PhD Thesis. Edmonton Alberta, Canada: University of Alberta; 2016. <https://doi.org/10.7939/R3KH0F69Q>
24. Lim HW, Song J. Efficient risk assessment of lifeline networks under spatially correlated ground motions using selective recursive decomposition algorithm. *Earthq Eng Struct Dyn*. 2012;41(13):1861-1882.
25. Valiant LG. The complexity of enumeration and reliability problems. *SIAM J Comput*. 1979;8(3):410-421.
26. Yoo YB, Deo N. A comparison of algorithms for terminal-pair reliability. *IEEE Trans Reliab*. 1988;37(2):210-215.
27. Frank H, Hakimi S. Probabilistic flows through a communication network. *IEEE Trans Circuit Theory*. 1965;12(3):413-414.
28. Evans JR. Maximum flow in probabilistic graphs—the discrete case. *Networks*. 1976;6(2):161-183.
29. Zhao P, Zhang X. A survey on reliability evaluation of stochastic-flow networks in terms of minimal paths. In: 2009 International Conference on Information Engineering and Computer Science. Wuhan, China: IEEE; 2009;1-4. <https://doi.org/10.1109/ICIECS.2009.5365424>
30. Esfahanian AH. *Connectivity Algorithms*. In: Beineke LW, Wilson RJ, eds. Topics in structural graph theory, Ch 12. New York: Cambridge University Press; 2013:268-281.
31. Ahuja RK, Magnanti TL, Orlin JB. *Network flows: Theory, Algorithms and Applications*. Los Angeles, California, United States: Prentice Hall; 1993.
32. Lee II EE, Mitchell JE, Wallace WA. Restoration of services in interdependent infrastructure systems: a network flows approach. *IEEE Trans Syst Man Cybern Part C Appl Rev*. 2007;37(6):1303-1317.
33. González AD, Dueñas-Osorio L, Sánchez-Silva M, Medaglia AL. The interdependent network design problem for optimal infrastructure system restoration. *Comput Aided Civ Inf Eng*. 2016;31(5):334-350.
34. Paredes R, Duenas-Osorio L. A time-dependent seismic resilience analysis approach for networked lifelines. In: Proceedings of ICASP12 12th International Conference on Applications of Statistics and Probability in Civil Engineering held in Vancouver; 2015; Vancouver, Canada. 1-8.
35. Vielma JP. Mixed integer linear programming formulation techniques. *SIAM Rev*. 2015;57(1):3-57.
36. Gurobi Optimization I. Gurobi optimizer reference manual; 2016.
37. Fishman GS. A Monte Carlo sampling plan for estimating network reliability. *Oper Res*. 1986;34(4):581-594.
38. Hernandez-Fajardo I, Dueñas-Osorio L. Probabilistic study of cascading failures in complex interdependent lifeline systems. *Reliab Eng Syst Saf*. 2013;111:260-272.
39. Dueñas-Osorio L, Craig JJ, Goodno BJ. Seismic response of critical interdependent networks. *Earthq Eng Struct Dyn*. 2007;36(2):285-306.
40. HAZUS. Multi-hazard loss estimation methodology, earthquake model, HAZUS-MH MR4 technical manual. *Technical Report*. Washington, D.C.; 2003.
41. González AD, Sánchez-Silva M, Dueñas-Osorio L, Medaglia AL. Mitigation strategies for lifeline systems based on the interdependent network design problem. In: International Conference on Vulnerability and Risk Analysis and Management (ICVRAM). Liverpool, England: American Society of Civil Engineers; 2014:1-10.
42. Field EH, Jordan TH, Cornell CA. OpenSHA: a developing community-modeling environment for seismic hazard analysis. *Seismol Res Lett*. 2003;74(4):406-419.
43. Field EH, Dawson TE, Felzer KR, et al. Uniform california earthquake rupture forecast, version 2 (UCERF 2). *Bull Seismol Soc Am*. 2009;99(4):2053-2107.
44. Boore DM, Atkinson GM. Ground-motion prediction equations for the average horizontal component of PGA, PGV, and 5PSA at spectral periods between 0.01 s and 10.0 s. *Earthq Spectra*. 2008;24(1):99-138.
45. Karger DR. A Fast and simple unbiased estimator for network (un)reliability. In: 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS); 2016; New Brunswick, NJ. 635-644. <https://doi.org/10.1109/FOCS.2016.96>
46. Duenas-Osorio L, Meel KS, Paredes R, Vardi MY. Counting-based reliability estimation for power-transmission grids. In: Proceedings of AAAI Conference on Artificial Intelligence; 2017; San Francisco. 4488-4494.

How to cite this article: Paredes R, Dueñas-Osorio L, Hernandez-Fajardo I. Decomposition algorithms for system reliability estimation with applications to interdependent lifeline networks. *Earthquake Engng Struct Dyn*. 2018;1–20. <https://doi.org/10.1002/eqe.3071>