ELSEVIER

Optical Switching and Networking 00 (2026) 1–

OSN

Hardness Results and an Efficient Heuristic for Safest Path Computation under Correlated Failures in Real-World and Synthetic Backbone Networks

Balázs Vass[a,b], Péter Revisnyei[a], Alija Pašić[c]

[a]*Department of Telecommunications and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3., Budapest, H-1111, Hungary*

[b]*Department of Mathematics and Computer Science of the Hungarian Line, Faculty of Mathematics and Computer Science, Babeș-Bolyai University, 1 Mihail Kogălniceanu Street, Cluj Napoca, RO-400084, Romania*

[c]*Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3., Budapest, H-1111, Hungary*

Abstract

To ensure a comprehensive evaluation and optimization of the expected availability of a backbone network service, two related fundamental modeling and algorithmic questions must be addressed. 1) Realistically estimate the availability of a given *st*-path for some source and target pair of communicating nodes *s* and *t*, and 2) Find a safest *st*-path. For these, traditional approaches suppose network element failures are independent. In this paper, it is demonstrated that, by not considering joint failure probabilities, the traditional approaches may misguess the *st*-path availabilities and, consequently, the total connection availability, which can lead to more frequent Service Level Agreement (SLA) violations and a financial burden on the Communication Service Provider (CSP). We show that the underlying problem is not only $\mathcal{NP}$-hard, but it does not admit a Fully Polynomial Time Approximation Scheme (FPTAS) either, unless $\mathcal{P} = \mathcal{NP}$. On the positive side, we propose a fast algorithmic approach, that, under some assumptions, returns with a (truly) safest st-path accompanied by its exact expected availability. When the aforementioned assumptions do not hold, in our simulations, the proposed algorithmic scheme proved to be a good heuristic, performing at least as well as the traditional method in the vast majority of cases. To provide further validation for the approach a synthetic network generation framework was introduced. It combines geographic constraints, minimum spanning tree–based connectivity, and Perlin-noise-driven landmass topology generation. This framework provides a range of realistic network scenarios for simulations, thereby ensuring the robustness of the evaluation. Across both real and synthetically generated networks, we observed the same consistent results, confirming the effectiveness of our method.

Keywords: survivable routing, safest path computation, disaster resilience, network failure modeling, probabilistic shared risk link groups, seismic hazard, synthetic networks, line graph, edge dual graph, inapproximability

1. Introduction

Many mission-critical services rely on uninterrupted network connectivity, whose robustness is commonly assessed via their Quality of Resilience (QoR) [2]. Examples include telesurgery and high-frequency trading systems, where extremely high reliability and availability are indispensable. Achieving such stringent performance levels depends not only on the underlying network infrastructure, but also on accurate failure modeling and the deployment of resilient routing strategies (e.g., protection mechanisms). Yet, most operational networks are designed to withstand only single link or node failures [3], or in some cases dual failures [4], which is insufficient for these demanding applications. Consequently, more rigorous Service Level Agreements (SLAs) are needed to satisfy the requirements of these critical communication services, whose importance continues to grow for both governments and the general public. An SLA constitutes a formal contract between a provider and a subscriber, specifying Quality of Service (QoS) parameters that capture the technical conditions the service must meet [5, 6]. Typical QoS metrics include packet loss, packet delay, guaranteed throughput, and port availability [7]. In addition, the concept of *service availability* was introduced in 2002 to quantify the fraction of time a service remains accessible to its users [7]. Violations of the agreed-upon service availability may incur monetary penalties for the communication service provider (CSP), in some cases reaching up to 15% of its Quarterly Payment [8]. Thus, to meet availability targets

Email addresses: balazs.vass@ubbcluj.ro (Balázs Vass), revisnyei@tmit.bme.hu (Péter Revisnyei), palija@hit.bme.hu (Alija Pašić)

An earlier version of this paper appeared at IEEE RNDM [1].

and avoid penalties, CSPs must carefully evaluate and improve the availability of their services, particularly for the primary paths that carry customer traffic. This requires accounting for single, independent, and correlated multi-element failures, including those triggered by natural disasters. In order to deliver high QoR at minimal operational cost, CSPs must therefore accurately estimate and subsequently optimize the availability of any given source–destination connection.

When it comes to a comprehensive availability estimation, there are two key components to consider:

1. *Modeling:* translating the essence of the complex real-world hazards that cause network element failures to concise combinatorial datasets.

2. *Computing:* using these datasets, for any given node pair s and t, to determine a reliable connection between them. This connection can be a safest st-path, st-path pair, etc.

Although recent works [9, 10, 11] have achieved significant advances regarding the *Modeling* component, to the best of our knowledge a comprehensive investigation of how to efficiently compute the safest st-connection (*Computing* component) has not yet been conducted. In this paper, we take initial steps toward addressing this latter problem. Our experiments show that the proposed fast algorithm for determining a safest st-path surpasses the traditional method, typically yielding paths that are at least as safe as those obtained by the classical approach.

1.1. Related Works

The computation of availability under *independent* single-element failures has been thoroughly investigated in the literature [12, 13, 14, 15, 16, 17]. Moreover, the treatment of correlated failures through *Shared Risk Groups (SRGs)* also has a long-standing research tradition [12, 13, 14, 15, 18, 19]. In this setting, an SRG typically refers to a small subset of links or nodes that exhibit a high likelihood of failing simultaneously. A central observation is that in many practical scenarios, like availability evaluation, the failure of a node v has the same effect as the concurrent failure of all links incident to v. In such cases, it is sufficient to operate with an appropriate collection of Shared Risk *Link* Groups (SRLGs), where every SRLG consists solely of links. Probabilistic extensions of the SRLG concept have also appeared in earlier work [20, 21, 9, 10, 11]. In particular, [11] established a unified and precise terminology for Probabilistic SRLGs (PSRLGs), which we adopt throughout this paper.

Routing under PSRLGs has also been studied before. E.g., papers[20, 21] are related to ours in their use of PSRLG information, but they address different objectives: [20] considers diverse routing under probabilistic failures, and [21] studies differentiated quality-of-protection provisioning. In contrast, we study the exact availability evaluation and optimization of a single (primary) st-path under PSRLG-based joint link-failure probabilities.

A natural strategy is to treat the disaster scenarios as part of the input [22], assuming that they have been carefully precomputed, for instance using historical hazard information. In contrast, much of the existing literature on regional failures has taken a less systematic route, employing problem-specific heuristics to model disasters in the context of network planning. Representative examples include identifying the most vulnerable regions of a network [23, 24, 25, 26], assessing the network-wide impact of a randomly occurring disaster [27, 28, 29], or quantifying by topological measures how well a network recovers from the damage [30]. Further works develop (re)routing and protection mechanisms that keep the traffic alive through a prescribed set of failures [31, 32, 33, 34, 35]. A separate line takes the infrastructure itself as the object of design, building resilience considerations into the network [36, 37, 38, 39].

At present, the work of [11] provides efficient techniques for representing and computing link–failure correlations, avoiding the need to either restrict the disaster set to a few scenarios or to inaccurately assume independence among link failures [40, 41, 42]. These techniques have already been integrated into sophisticated frameworks for disaster-aware resilience [43, 44]. Nevertheless, to the best of our knowledge, the problem of efficiently determining a (truly) safest st-path has remained unresolved.

Making steps towards the solving of this problem, we make use of the so-called *line-graph*s [45], which are a standard tool whenever path costs depend on link sets, each set being incident to a given node. In this regard, our contribution is not the pure use of the line-graph transformation by itself. Instead, vaguely speaking, our observation is that, under upcoming Property 1, the exact formula for path availability reduces to terms associated with single links and consecutive link pairs, thus, these terms can be represented as nonnegative edge costs in an auxiliary graph (resembling the line-graph), so that a shortest path in the auxiliary graph corresponds exactly to a safest path in the original network. To the best of our knowledge, no previous study leveraged line graphs for PSRLG-based availability optimization.

Synthetic network generation has a long-standing history and it has been widely employed to construct controlled benchmarking environments. Since over the years, numerous topology generation models and frameworks have been proposed, the following examples do not constitute an exhaustive overview. Among the earliest approaches, the Waxman topology generator [46] focused primarily on the generation of general random network structures. Subsequently the Georgia Tech Internetwork Topology Models (GT-ITM) [47, 48] emphasized the hierarchical organization of Internet topologies. The Boston University Representative Internet Topology gEnerator (BRITE) [49] was developed to reproduce Internet-like connectivity characteristics and routing-related structural properties. Another framework, Tiers [50], introduced a three-level hierarchical generation methodology intended to emulate the distinction between wide-area, metropolitan-area, and local-area networks within the Internet infrastructure. More recently, Soltan et al. [51] proposed the Geographical Network Learner and Generator (GNLG) algorithm for synthesizing power grid networks that preserve the structural characteristics of a reference system. Nevertheless, none of the aforementioned tools

integrate adequately into our technical pipeline or satisfy the specific requirements of our application domain. Consequently, these models primarily serve as conceptual and methodological inspiration for the development of our proposed approach.

1.2. Main Contributions

Motivated by the above, our paper delves into the study of the correctness and efficiency of some safest path computation algorithms. Also, this paper aims to advance the theoretical foundations and evaluation of our proposed method in [1]. The main contributions of the paper are summarized as follows:

- We show that, by not considering joint failure probabilities, the traditional approaches may significantly misestimate the total connection availability, which can lead to more frequent SLA violations and a financial burden on the CSP. Due to this inconsistent (over)estimation, the supposedly safest paths yielded by these approaches may turn out to be suboptimal.
- We show that the problem investigated is not only $\mathcal{NP}$-hard, but it does not admit a Fully Polynomial Time Approximation Scheme (FPTAS) either, unless $\mathcal{P} = \mathcal{NP}$.
- We propose a fast algorithmic approach, that, for a graph $G = (V, E)$, under some realistic assumptions, returns with a (truly) safest st-path accompanied by its exact expected availability in $O(|V| \log^{2/3} |V|)$.
- We introduce a novel framework for synthetic network generation that can produce realistic network topologies and earthquake disaster scenarios.
- We provide a proof of concept implementation and simulations based on both real-world and synthetic network topologies with seismic hazard data. Our evaluations show that, in the general case, our proposed method is an efficient heuristic in guessing a safest path, outperforming the traditional approach.

Among the main contributions The following parts constitute the primary extensions over the conference version [1]:

1. Proving that the problem at hand does not admit a FPTAS, unless $\mathcal{P} = \mathcal{NP}$, extends the earlier complexity results and clarifies the inherent computational limitations of the problem.
2. Introducing a novel synthetic network generation framework.
3. Extending our evaluations we used the synthetic networks to demonstrate the robustness, scalability, and practical relevance of the proposed approaches.

Moreover the entire text has been thoroughly revised, clarified, and we enhanced the structure of the algorithmic arguments. Thus, the results are clearer, more grounded, and more complete in this paper.

The rest of the paper is organized as follows. Section 2 introduces our formal model and discusses some basic results. In Section 3, a fast algorithm is described that, under certain circumstances, is guaranteed to compute a safest st-path. Further, our simulation results for the real-world networks are presented in Section 4. Next in Section 5, the process of the synthetic network generation and the results based on these networks are discussed. In Section 6, the results of the real-world and synthetic networks are compared. Finally, Section 7 concludes our paper.

2. Model and Basic Results

The problem input consists of two parts. One is a connected graph $G = (V, E)$ with vertex set V ($|V| \geq 2$) and edge set E, along with a communication source-target node pair $\{s, t\} \subseteq V$.

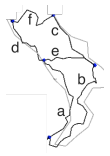
The other part of the problem input encodes the joint failure probabilities of link sets. For this, for a link set $S \subseteq E$, in line with [11, 44], we define CFP(S) ('cumulative failure probability of S') to denote the probability that *at least* link set S (and maybe some other links too) will fail at the next disaster. The second part of the input is CFP[G], which is a data structure containing all the CFP(S) values, where we list CFP(S) only if CFP(S) > 0. Note that in most of the natural settings, CFP[G] has a manageable size [52]. Also, while the input CFP[G] focuses only on common failures of *links*, if necessary, these structures can store failure probabilities of both links *and* node failures (see [11, Sec. V.]).

The goal is to find a safest st-path, i.e., a path between nodes s and t that has a lowest chance that any of its links will fail under the next disaster. The decision version of the problem can be formalized as Problem 1.

The joint failure probabilities of network elements could be represented in other ways, too. To show that the hardness of

Input: network $G = (V, E)$, nodes $\{s, t\} \subseteq V$, and CFP[G] consisting of set-probability pairs S, CFP(S) for $S \subseteq E$: CFP(S) > 0, where CFP(S) denotes the probability that *at least* S will fail during the next disaster

Network G:



CFP[G] :

CFP(a) $=4.07_{10^{-2}}$ CFP(b) $=3.53_{10^{-2}}$ CFP(a,b) $=5.68_{10^{-3}}$ CFP(b,e) $=6.91_{10^{-6}}$ CFP(a,d,e) $=3.27_{10^{-4}}$ CFP(a,b,e) $=0$

CFP(c) $=1.13_{10^{-2}}$ CFP(d) $=2.91_{10^{-3}}$ CFP(a,e) $=4.59_{10^{-4}}$ CFP(c,e) $=7.48_{10^{-4}}$ CFP(b,c,e) $=6.91_{10^{-6}}$

CFP(e) $=1.46_{10^{-2}}$ CFP(f) $=2.60_{10^{-2}}$ CFP(d,e) $=3.27_{10^{-4}}$ CFP(d,f) $=2.78_{10^{-4}}$

CFP(c,f) $=5.25_{10^{-4}}$ CFP(b,c) $=7.27_{10^{-6}}$

CFP(a,d) $=3.35_{10^{-4}}$

Output: a safest st-path

Figure 1. Example real-world problem input. The depicted network G is a part of the Interoute network [10] covering the southern part of continental Italy. Joint failure probabilities due to earthquakes are also taken from [10], calculated at a shaking intensity tolerance of VI. In this example, Property 1 holds.

Problem 1: Safest Path CFP Problem - decision version

Input: A graph $G = (V, E)$, nodes s and t, a threshold T, and failure probabilities CFP[G].

Question: Decide whether there exists an st-path that, after the next disaster, remains intact with a probability of at least T. If such a path exists, the instance is *satisfiable*.

Problem 1 (see Sec. 2.2) is not coming from an unfortunate representation of the failure data, in the following, we phrase the same problem, but instead of using the CFP values, we use a data set consisting of FP(S) values, where FP(S) (*link failure state probability of* S) denotes the probability that, under the next disaster, the failed link set will be exactly S. With the FP values, the problem at hand is formalized as follows.

Problem 2: Safest Path FP Problem - decision version

Input: A graph $G = (V, E)$, nodes s and t, a threshold T, and failure probabilities FP[G].

Question: Decide whether there exists an st-path that, after the next disaster, remains intact with a probability of at least T. If such a path exists, the instance is *satisfiable*.

Throughout the theoretical reasoning, to get closer to the practical time complexities in real-world backbone networks, we use the following parameters. Δ is the maximal degree of a node in G (that is often $\leq$ 4). We denote the number of link crossings by x (typically, $x \ll |V|$ because, usually, at crossings, optical cross-connects (OXCs) are deployed). For a path P, the maximal number of network links that a disaster hitting P hits in G is denoted by ρ_P (i.e., $\rho_P := \max\{S \subseteq E \mid \text{CFP}(S) > 0, \text{ and } S \cap P \neq \emptyset\}$). Lastly, the length of a path P is sometimes denoted by k.

We note upfront that our fast algorithm to be presented in Sec. 3 is guaranteed to return with a safest st-path if the following Property holds:

Property 1. *Any set in CFP*[G] *is star-like, i.e., each of these sets is a subset of links incident to a network node* $v \in V$.

In certain real-world scenarios, the CFPs related to a network, and encoding the hazards of some natural disasters, fulfill Property 1. E.g., see Fig. 1, where, for a part of the Interoute network [10], the CFPs representing the seismic hazard are depicted. When Property 1 does not hold, our algorithm can be viewed as a fast heuristic for finding safe st-paths.

Finally, we note that, in the complexity analysis, we use the unit cost arithmetic model of computation, where the cost of a basic operation with real numbers is 1. The basic operations allowed here are $+, -, \times, /$. The rationale behind the use of the unit cost model is that the complexity stemming from arithmetic operations in this model is negligible.

Next, we present the standard traditional trick for finding a (supposedly) safest st-path in Sec. 2.1, then, we discuss exact methods for path availability computation in Sec. 2.3.

2.1. Standard method for safest paths and its limitations

The standard method for finding a supposedly safest st-path that unrealistically assumes that network element failures are independent goes as follows. For each link $e \in E$, we will assume that the probability that e fails is less than 1, i.e., $0 \leq \text{CFP}(e) < 1$ (otherwise, the always-failing link can be deleted from the network). By assuming the independence of the link failures, the availability $A_{\text{indep}}(P_{st})$ of an st-path P_{st} can be expressed as

$$A_{\text{indep}}(P_{st}) = \Pi_{e \in P}(1 - \text{CFP}(e)). \tag{1}$$

The following trick helps us to find a path that maximizes the above expression. We take the logarithm of the RHS of Eq. (1). Since the logarithm is a strictly monotone function, the maximum of the new expression will be reached on the same path(s) that also maximize Eq. (1). Since the logarithm of a product is the sum of the logarithms, it is enough to find a path maximizing $\sum_{e \in P} \log(1 - \text{CFP}(e))$. Luckily, this is equivalent of finding a path minimizing $\sum_{e \in P} -\log(1 - \text{CFP}(e))$. This can be done by e.g., the Dijkstra algorithm [53], since $-\log(1 - \text{CFP}(e)) > 0$ for all $e \in E$.

It is easy to see that this approach cannot compute the exact availability of a path, thus it is not suitable for determining a safest st-path. To give an oversimplified example, suppose we have two st-paths P_1 and P_2 made up of links $\{x, y\}$, and $\{z\}$, respectively. Let $\text{CFP}(x) = \text{CFP}(y) = 0.5$, $\text{CFP}(z) = 0.9$, while the rest of the link sets have CFPs of zero (including $\text{CFP}(x, y) = 0$). Then, the availability of path P_1 estimated by the standard method is just 0.25, which is far better than the 0.1 reached by P_2. Thus, the standard method would recommend using path P_1. In reality, however, this is a terrible idea, since, with probability 1, either link x or y of path P_1 is not working, thus its real availability $A(P_1) = 1 - \text{CFP}(x) - \text{CFP}(y) + \text{CFP}(x, y) = 0$. Therefore it is better to use path P_2.

Turning to a simple real-world example, Fig. 1 depicts a part of the Italian telecom network Interoute [10] accompanied by CFPs computed based on the seismic hazards. There, the probability that route $f-d-e$ remains operational after the next earthquake is calculated as $\approx .9602$ by the standard approach, compared to the real availability of $\approx .9570$. This means a 7.7% underestimation of the unavailability.

The next subsection will show that finding a safest path is an $\mathcal{NP}$-hard and inapproximable problem. This hardness result partly explains why such heuristic mathods like the one just detailed are being used for determining a supposodly safest path.

2.2. $\mathcal{NP}$-hardness and inapproximability

Lately, [54] proved that optimizing the availability of a path P is $\mathcal{NP}$-hard. In the following, by slightly modifying and adding some observations to the proof given by [54], we prove a stronger result, namely, the inapproximability of the problem.

Theorem 1. *For Problem 1 and 2, respectively, there is no Fully Polynomial Time Approximation Scheme (FPTAS), unless* $\mathcal{P} = \mathcal{NP}$. *More precisely, no polynomial-time* $1 - \frac{15}{4|V|}$ *approximation exists for these problems, unless* $\mathcal{P} = \mathcal{NP}$.

Proof:. In the inapproximability proof, we will refer to the $\mathcal{NP}$-hard problem `(3,4)-SAT` [55, Theorem 2.1.], where instances are 3-CNF formulas where each clause has exactly 3 literals and each variable appears in at most 4 clauses (counting both positive and negative occurrences). Denoting the number of variables by ρ, and the number of clauses by k, it is trivial that in any `(3,4)-SAT` problem instance, $k \leq 4\rho$.

The construction for reduction we detail in the following is essentially the same as in [54], that provided a polynomial-time reduction from the $\mathcal{NP}$-complete `3-SAT` problem [56] to both Problem 1 and Problem 2. The construction goes as follows.

First, we will provide a polynomial-time reduction from the $\mathcal{NP}$-hard `(3,4)-SAT` problem [55, Theorem 2.1.] to (the decision variant of) the `Safest Path FP` problem. Inapproximability of `Safest Path CFP` will be proved via an extra observation at the end of this proof.

Suppose we are given a `(3,4)-SAT` problem instance f, that is a conjunctive normal form, standing of clauses containing exactly three literals each, and each variable appearing at most 4 times in the instance. That is, $f = C_1 \wedge C_2 \wedge \cdots \wedge C_k$, with each C_i ($i \in \{1, \ldots, k\}$) equalling $(l_{i,1} \vee l_{i,2} \vee l_{i,3})$. The `(3,4)-SAT` problem is to determine whether f is satisfiable. In the following, we construct a `Safest Path FP` problem instance that is satisfiable exactly if f is satisfiable and has an input of size $O(k)$ that can be computed from f in $O(k)$.

Going into details, known formula f that has k clauses, and ρ variables, we define graph $G = (V, E)$ as follows. Let the set of nodes be

$$V := \{v_{i,j} \mid i \in \{1,2,3\}, j \in \{1, ..., k+\rho+1\}\} \cup \{s, t\}.$$

Additionally, the set of edges is defined as follows:

$$E := \{\{v_{1,j}, v_{2,j}\}, \{v_{2,j}, v_{3,j}\} \mid j \in \{1, k+\rho+1\}\} \cup$$

$$\cup \{\{v_{i,j}, v_{i,j+1}\} \mid i \in \{1,2,3\}, j \in \{1, \ldots, k+\rho\}\} \cup \{\{s, v_{1,1}\},$$

$$\{s, v_{2,1}\}, \{s, v_{3,1}\}, \{v_{1,k+\rho+1}, t\}, \{v_{2,k+\rho+1}, t\}, \{v_{3,k+\rho+1}, t\}\}.$$

Intuitively, G can be drawn as a $3 \times (k+\rho)$ grid graph, with s and t appended to the nodes on the first and last columns. Intuitively, the links that may fail at the next disaster are exactly the horizontal ones, as explained in the following.

Each link set S that has a positive failure probability ($\mathrm{FP}(S) > 0$) is set to have a failure probability $\mathrm{FP}(S) = \frac{1}{2\rho}$. For each variable x_σ ($\sigma \in \{1, \ldots, \rho\}$), we define two such link sets (this makes a total of 2ρ link sets of positive failure probability; thus the failure probabilities add up to 1 as expected). The first set, S_σ contains each link $\{v_{b,\sigma}, v_{b,\sigma+1}\}$ ($b \in \{1,2,3\}, \sigma \in \{1, \ldots, \rho\}$) for which, in the σ^{th} clause, the b^{th} literal is x_σ (as a positive literal); and, in addition, S_σ contains link $\{v_{1,k+\sigma}, v_{1,k+\sigma+1}\}$. The second link set associated to x_σ, denoted by $S_{\neg\sigma}$, contains each link $\{v_{b,\sigma}, v_{b,\sigma+1}\}$ ($b \in \{1,2,3\}, \sigma \in \{1, \ldots, \rho\}$) for which in the σ^{th} clause the b^{th} literal is $\neg x_\sigma$ (as a negative literal); and, in addition, $S_{\neg\sigma}$ contains links $\{v_{2,k+\sigma}, v_{2,k+\sigma+1}\}$ and $\{v_{3,k+\sigma}, v_{3,k+\sigma+1}\}$.

The probability threshold T is set to $\frac{1}{2}$. Since any st-path P has to pass at least one of links $\{v_{i,k+\sigma}, v_{i,k+\sigma+1}\}$, $i \in \{1,2,3\}$, it has to intersect either S_σ or $S_{\neg\sigma}$. Thus, the survival probability of path P cannot be higher than $\frac{1}{2}$. Further, we can observe that the availability of P can be (at least) $\frac{1}{2}$ precisely if the `(3,4)-SAT` problem instance f is satisfiable.

In this construction, it is $\mathcal{NP}$-hard to whether the availability of exactly $1/2$ can be achieved. Also, due to the construction, if the availability of $1/2$ cannot be achieved, it can be at most $\frac{1}{2}(1-1/\rho)$. This means it is $\mathcal{NP}$-hard to approximate the optimum within a fraction that is bigger than $1 - 1/\rho$. The negative term can be underestimated as follows:

$$\frac{1}{\rho} = \frac{1}{\frac{1}{5}(\rho + 4\rho)} \geq \frac{5}{4(\rho + k)} =$$

$$\frac{15}{4}\frac{1}{3(\rho+k)} \geq \frac{15}{4}\frac{1}{3(\rho+k)+5} = \frac{15}{4} \cdot \frac{1}{|V|}.$$

Thus, the inapproximability ratio for Problem 2 being $1 - 1/\rho$ can be upper estimated by $1 - \frac{15}{4|V|}$,

In the construction presented above, for each S, $\mathrm{FP}(S) = \mathrm{CFP}(S)$, thus the same construction is also a proof of the claimed inapproximability ratio of `Safest Path CFP`. □

2.3. Evaluating the availability of a path

Despite the inherent hardness of our *optimization* problem, evaluating the availability of a path is relatively straightforward. The conference version of the paper described ways to compute the availability of a path given the CFP values collected in CFP[G]. In the following, we describe a conceptually simpler method for availability evaluation, which is based on the link failure probability $\mathrm{FP}(S)$ of each link set S. Recall that $\mathrm{FP}(S)$ denotes the probability that, under the next disaster, the failed link set will be exactly S. Known the graph G, and the list FP[G] of link sets of positive FP, there is a straightforward way to evaluate the *real* availability of $A(P)$ a given st-path P. Namely, a path P fails if a failure event cuts at least one link on the path:

$$A(P) = 1 - \sum_{S \cap P \neq \emptyset} \mathrm{FP}(S). \tag{2}$$

Note that to evaluate the above expression, only the link sets S being part of FP[G] should be considered, the FP value of other link sets is zero by definition

Alternatively, using the CFP values, the exact availability of a path P can be obtained by applying the principle of inclusion–exclusion [57]. It results in the following expression:

$$A(P) = 1 + \sum_{S \subseteq P} (-1)^{|S|} \mathrm{CFP}(S). \tag{3}$$

Indeed, defining $\mathrm{CFP}(\emptyset) = 1$ for a moment, we have:

$$A(P) = \Pr[F \cap P = \emptyset] = \sum_{S \subseteq P} (-1)^{|S|} \Pr[S \subseteq F] = \sum_{S \subseteq P} (-1)^{|S|} \mathrm{CFP}(S).$$

3. A fast algorithm for finding a safest st-path

In 2.1, we saw that computing a safest st-path is easy if link failures are supposed to be independent. Naturally arises the

question of whether there is a broader class of inputs that can be solved efficiently. In this Section, we will suppose that Property 1 holds. For this case, we show a fast algorithm that, under some practical assumptions, and pulling in the shortest-path finding algorithm of [58] as a subroutine, achieves a worst-case time complexity of $O(|V| \log^{2/3} |V|)$.

Our first observation is that, in presence of Property 1, Eq. (3) becomes much simpler:

Claim 1. *Supposing Property 1, the availability of a path $P = \{e_1, \ldots, e_i\}$ can be calculated as:*

$$A(P) = 1 - \sum_{j=1}^{i} CFP(\{e_i\}) + \sum_{j=1}^{i-1} CFP(\{e_j, e_{j+1}\}) \qquad (4)$$

Proof:. On the beginning of the proof, we record the simple structural fact behind Property 1. Let S be a star-like link set incident to node v, and let P be a simple path. Then P can contain at most two links of S. Indeed, indirectly, $|P \cap S| \geq 3$ would mean at least three links of path P incident to node v, that is a contradiction. Moreover, if P contains two links of S, then these two links are consecutive on P.

By Property 1, every link set S with $\mathrm{CFP}(S) > 0$ stands of links incident to a node v. By the preceding observation, for any simple path P, the intersection $S \cap P$ is either empty, a singleton, or a pair of consecutive links of P. Consequently, in the inclusion– exclusion expression Eq. (3), the only nonzero terms involving subsets of P are the singleton terms and the terms corresponding to consecutive link pairs of P. This yields Eq. (4). □

Second, instead of maximizing the availability, we will minimize the unavailability of the *st*-path. The unavailability of path P is defined simply as $U(P) := 1 - A(P)$. Thus, supposing Property 1, we have:

$$U(P = \{e_1, \ldots, e_i\}) := \\ = \sum_{j=1}^{i} \mathrm{CFP}(\{e_i\}) - \sum_{j=1}^{i-1} \mathrm{CFP}(\{e_j, e_{j+1}\}). \qquad (5)$$

Thirdly, instead of minimizing on graph G itself, we will minimize the unavailability in a graph derived from G (see Figure 3), which is very similar to the so-called line graph of G [45]. We start by describing the line graph.

Definition 1 (Line graph). *A line graph $L(G) = (L(V), L(E))$ (also called edge-to-vertex dual) of a graph G is obtained by associating a vertex with each edge of the graph and connecting two vertices with an edge if and only if the corresponding edges of G have a vertex in common. For an edge $e \in E$, we denote the node in $L(V)$ corresponding to e as e'.*

To get the graph we need for easily computing a safest *st*-path, we need to modify $L(G)$, intuitively speaking, around s and t. We will call the resulting graph as *'edge dual'* graph.

Definition 2 (Edge dual). *The st-edge dual graph $G'_{st} = (V'_{st}, E'_{st})$ of graph $G = (V, E)$ is derived from the line graph $L(G) = (V(G), E(G))$ as follows. We add nodes s and t to $V(G)$, i.e., $V'_{st} := V(G) \cup \{s, t\}$. Also, for each edge e incident to s and t in G, we add edge $\{s, l(e)\}$ and $\{t, l(e)\}$, respectively, where $l(e) \in V(G)$ denotes the node in $L(G)$ corresponding to link e in G. More formally, $E'_{st} := E(G) \cup \{\{s, l(e)\} | e = \{s, v\} \in E\} \cup \{\{t, l(e)\} | e = \{t, v\} \in E\}$.*

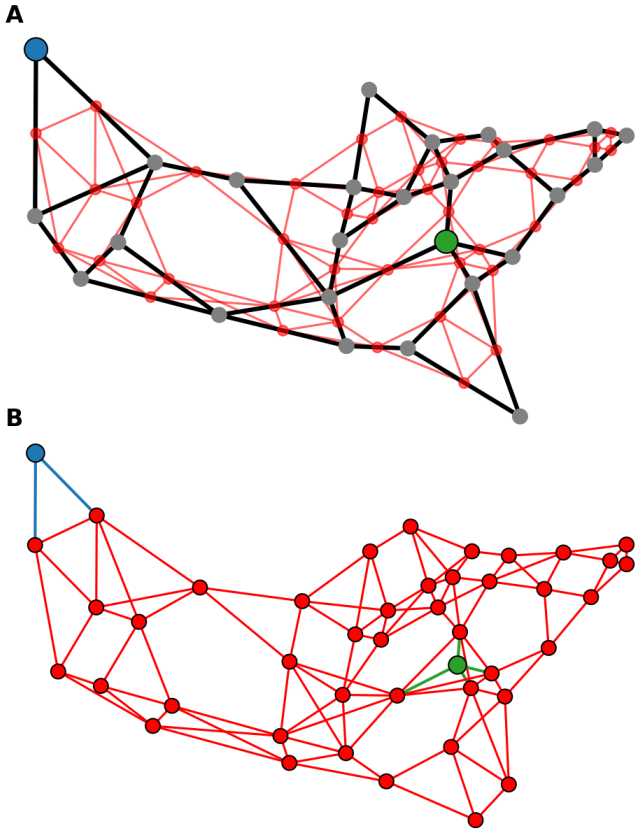


Figure 2. Illustration of the edge-dual transformation on the real-world network `janos-us`. **A -** The original graph G is shown (gray nodes, black edges) with the chosen source node s (blue) and target node t (green). The line graph $L(G)$ is overlaid dimly (red), where each edge of G becomes a node of $L(G)$. **B -** The *st*-edge dual graph G'_{st}, obtained from $L(G)$ by adding node s, node t and extra edges (blue and green) to $L(G)$. These edges connects the s and t nodes to the line-graph nodes which represents the links of G incident to s and t, respectively.

Figure 2 illustrates the line graph and edge-dual transformation of a network for a specific *st* pair. Figure 2/A shows the original graph G (gray nodes and black edges) with a chosen s (blue) and t (green) node. The first step of the transformation is the edge-to-vertex conversion (Def. 1) which results in the line graph which is also presented there (red graph). The second step is expanding the line graph with the original s and t nodes and the additional edges (blue and green) connected to these nodes. This step creates the edge dual graph (Def. 2) which is shown on Figure 2/B.

When it does not cause confusion, we refer to the *st*-edge dual simply as the edge dual, and simplify its notation $G'_{st} = (V'_{st}, E'_{st})$ to $G' = (V', E')$.

Finally, we define a cost function on the edges of G'.

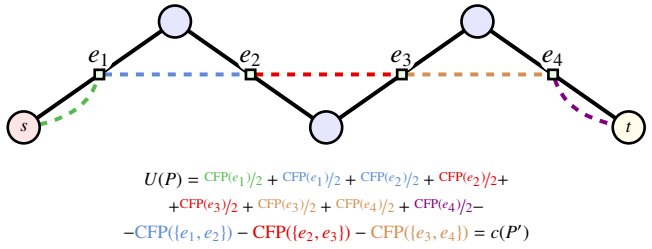


Figure 3. Illustration of an st-path $P = \{e_1, e_2, e_3, e_4\}$ (solid links) and its counterpart P' (Def. 4, dashed links) in the st-edge dual graph (Def. 2). If Property 1 holds, $U(P) = c(P')$ (Claim 2); cost c (described in Def. 3) is nonnegative, thus a cheapest (most reliable) st-path can be calculated via a simple Dijkstra.

Definition 3 (Edge dual cost function). *Based on graph G, and the collection of CFPs CFP[G], the edge dual cost function $c : E' \to \mathbb{R}^0_+$ assigns nonnegative costs to the edges in the edge dual, as described in the following. For links f, g adjacent in G, let the cost of link $\{f', g'\}$ in G' be $c(\{f', g'\}) := \frac{1}{2}CFP(f) + \frac{1}{2}CFP(g) - CFP(f, g)$. Let the cost of links $\{s', l(e)\}$ incident to s' be $c(\{s', e'\}) := \frac{1}{2}CFP(e)$. Similarly, let the cost of links $\{t', e'\}$ incident to t' be $c(\{t', l(e)\}) := \frac{1}{2}CFP(e)$.*

It is useful to define the following one-to-one correspondence between st-paths P graph G and their counterparts P' in the edge dual G'.

Definition 4. *For an st-path $P = e_1, \ldots, e_i$ in G, we define its* counterpart *as $P' := \{s', e'_1\}, \{e'_1, e'_2\}, \ldots, \{e'_{i-1}, e'_i\}, \{e'_i, t'\}$. We define the counterpart of P' to be P.*

Claim 2. *Given any st-path P in G, and its counterpart P' in G', and supposing Property 1 holds, $c(P') = U(P)$.*

Proof:. For an arbitrary st-path $P' = \{s', e'_1\}, \{e'_1, e'_2\}, \ldots, \{e'_{i-1}, e'_i\}, \{e'_i, t'\}$, we can derive the following chain of equations: $c(P') = c(\{s', e'_1\}) + c(\{e'_1, e'_2\}) + \cdots + c(\{e'_{i-1}, e'_i\}) + c(\{e'_i, t'\}) = \frac{1}{2}\text{CFP}(e_1) + \sum_{j=1}^{i-1}(\frac{1}{2}\text{CFP}(e_j) - \text{CFP}(e_j, e_{j+1})) + \frac{1}{2}\text{CFP}(e_i) + \frac{1}{2}\text{CFP}(e_i) = \sum_{j=1}^{i}\text{CFP}(e_i) - \sum_{j=1}^{i-1}\text{CFP}(e_j, e_{j+1}) = U(P)$. □

Corollary 1. *An st-path P is a safest st-path (i.e., $U(P)$ is minimal among the st-paths) exactly if its counterpart P' is a cheapest st-path in G'.*

Theorem 2. *Given an input graph G with the cumulative failure probabilities CFP[G], nodes s and t, and supposing Property 1 holds, a safest st-path P can be computed in $O(|E|^2 + C)$, or $O(\Delta(|V| + x) + C)$ worst-case time complexity, where C denotes the complexity of an arbitrary algorithm suitable for finding a shortest st-path in the edge dual G' and associated nonnegative edge dual cost function c.*

Specifically, using Dijkstra's algorithm, or the algorithm of [58], the resulting complexity is $O((|V| + x)(\Delta + \log|V|))$, and $O(\Delta(|V| + x)\log^{2/3}|V|)$, respectively.

Proof:. Due to Cor. 1, it is enough to search for a cheapest st-path in the st-edge dual graph G', since it counterpart will be a safest st-path in G. Since the cost function c is defined to be nonnegative, we may use a simple Dijkstra [53] algorithm to find such a path P'. Its best worst case time complexity is $O(m + n\log n)$ [59], where m and n are the number of edges and nodes in the graph, respectively. Alternatively, the algorithm of [58] takes $O(m\log^{2/3} n)$, that is faster on sparse graphs (i.e., for most edge dual graphs calculated from backbone topologies)[1].

Clearly, $G' = (V', E')$ has $|E| + 2$ nodes: one corresponding to each edge in G, and two for s and t, respectively. A trivial asymptotic upper bound for $|E'|$ is $O(|E|^2)$, since there could be an edge in E' for any link pair in E. In practice, for backbone topologies, the maximum node degree Δ is typically 'small'. Using parameter Δ, we can tell a $O(\Delta|E|)$ upper bound for $|E'|$. Also, by [19], $|E|$ is $O(|V| + x)$, where x denotes the number of link crossings in the network (and typically, $x \ll |V|$). With this, we get that $|E'|$ is $O(\Delta(|V| + x))$.

Building the st-edge dual takes $O(|E'|)$ as follows. V' can be built while scanning through the list of links in G once (and then adding s and t, respectively). Meanwhile, for each $v \in V$, the list $i(v)$ of links incident to v can be calculated. By adding the counterparts of edge pairs in each $i(v)$, and then adding the extra edges around s and t, $|E'|$ can be computed using $O(|E'|)$ additional operations.

Combining the above claims, we get the complexities proposed by the theorem. □

Corollary 2. *Since, in practice, for backbone networks Δ can be upper estimated by a small constant, and x is typically much smaller than $|V|$ (because at edge crossing points, typically an OXC is located, that counts as a node in our model), we can anticipate a typical runtime of $O(|V|\log|V|)$ for the runtime of a properly implemented Dijkstra on the edge dual. Substituting Dijkstra with the algorithm of [58], this complexity drops to $O(|V|\log^{2/3}|V|)$.*

Note that if one chooses Dijkstra's as the shortest-path finding algorithm, for a fixed source s, one may compute the tree of safest paths for all endpoints $t \in V$ in the same complexity of $O(|V|\log|V|)$. This is due to the combination of the properties of Dijkstra's and the bijection created between the original problem of finding safest paths of G and finding shortest paths on the edge dual of G.

4. Evaluation of results on real-world data

In this section, we assess the performance of the algorithms under consideration. Our simulations use real-world network topologies together with CFP values derived from actual earthquake disaster datasets (as computed in [11, 44]). We begin

[1] Technically, [58] tackles directed graphs, but it can be used in our case since the following. Although G' is defined as an undirected graph, any shortest-path algorithm for directed graphs with nonnegative arc costs can be applied without loss of generality by replacing every undirected edge $\{a, b\} \in E'$ with the two directed arcs (a, b) and (b, a), both having the same cost. This symmetric directed representation preserves shortest-path costs exactly.

by describing the simulation environment, after which we compare the resulting availability metrics. Subsequently, we analyze the performance of the two path-finding approaches and examine the relative lengths of the paths they produce. Finally, we report the runtimes of the various methods. The insights obtained from real-world networks are further underpinned by extensive experiments on synthetic topologies, which are presented in Section 5.

4.1. Simulation settings

The simulations in this study were conducted using five real-world communication networks. The nfsnet, janos-us, cost266 and optic-eu are obtained from [19], and the Interoute network is taken from [10]. These networks cover the territory of the USA, Europe, and Italy. In this paper, conform to [10], it is expected that network elements fail at a ground movement intensity threshold of VI.

The number of nodes, edges, the average internode distance, the average node degree and the average node connectivity are presented in Table 1. Also this table contains the simulation details for each network. For each network, we conducted the simulations for each possible *st* point pair. Thus, for each network, the number of simulations equals $\binom{|V|}{2}$.

In the simulations, two distinct methods were independently employed to find the safest path between the given s and t nodes. In both cases, the Dijkstra algorithm was utilized to identify the safest path, but the process of calculating the weight of the edges was different. In the *Independent method* (IM), weights were calculated by the standard algorithm, detailed in Sec. 2.1. In the *Edge-dual method* (EDM), a so-called edge-dual graph and its weights are calculated by our fast heuristic, as presented in Sec. 3. The safest paths derived from these methods are denoted as P_{indep} and P_{edm}.

Three different availability measures were used as evaluation metrics: for each path P,

(1) $A(P)$ is the (actual) availability defined in Eq. (3),

(2) $A_{\text{indep}}(P)$ is the availability estimation used IM, defined in Eq. (1), and finally,

(3) $A_{\text{ed}}(P)$ is the availability estimation used in EDM, defined in Eq. (4).

We note again that if Property 1 holds, $A(P) = A_{\text{ed}}(P)$ is true for any path P.

The simulations were conducted on a virtual machine equipped with an AMD Ryzen 5 2500U (8 cores @ 2.0GHz) processor and 16 GB of RAM. The simulation environment and the algorithms are implemented in Python 3.11.5.

4.2. Availability metrics comparison

In this subsection, the different availability metrics are compared by dividing each routing method's own availability metric (which is maximized in the algorithm) by the actual availability of a path.

For IM, the average ratio of $A_{\text{indep}}(P)$ to $A(P)$ is 0.999. In 4% of the simulations, this ratio exceeds 1, indicating that the $A(P)$ is smaller than the $A_{\text{indep}}(P)$, meaning A_{indep} overestimates the availability in these cases. For 12% of the simulations, the availabilities are equal. In the remaining 84% of the simulations, the $A_{\text{indep}}(P)$ is smaller, thus we can conclude that A_{indep} typically underestimates the availability of the determined paths.

In case of EDM, as discussed in Sec. 3, if Property 1 holds, $A_{\text{ed}}(P)$ and $A(P)$ are equal. Our simulations show that even without the fulfillment of Property 1, most of the times the differences between these values are zero or in the order of 10^{-16}, which is negligible.

4.3. Path availability comparison

In this subsection, we analyze the effectiveness of the IM and EDM by comparing the availability of the safest paths determined by the methods.

Table 1 presents a comparison of the availabilities of the paths yielded by the IM and EDM, respectively. It is clearly visible that most of the time, the availability of the paths provided by the different routing methods is equal. This occurs because they frequently identify the same path. However, in each network, there are *s*,*t* pairs where the availability of the path determined by EDM ($A(P_{\text{edm}})$) is higher than the availability of the path recommended by IM ($A(P_{\text{indep}})$). Conversely, in our simulations, the IM never provided a safer path than the EDM in the examined real-world networks.

Where $A(P_{\text{edm}})$ is higher, the average availability differences between respective *st*-path pairs are presented in Table 1.

The difference between the availabilities is the most significant in the case of the Interoute network with $1.76 \cdot 10^{-3}$. This is probably due to its relatively short links, and strong earthquakes. Thus, we put it in further perspective. Here, by choosing nodes s and t as can be seen on Fig. 4, we get $A(P_{\text{edm}}) - A(P_{\text{indep}}) \simeq 2.98 \cdot 10^{-3}$. In the following, we translate this value to yearly downtime. First of all, in fact, $A(P)$ denotes the probability that path P will fail *when the next disaster strikes*. In Italy, there is an expected number of $r = 5.53$ earthquakes that are considered (that have a strength of $> 4.5 M_w$) [11]. For the sake of estimation, we apply a Mean Time To Repair (MTTR) of 24 hours, equaling 1440 minutes [44] (this MTTR might be a slightly optimistic under-estimation in case of an earthquake). With this, the expected difference in the downtimes of paths P_{edm} and P_{indep} due to an earthquake can be calculated as follows:

$$\left(A\left(P_{\text{edm}}\right) - A\left(P_{\text{indep}}\right)\right) \cdot r \cdot \text{MTTR} \simeq 2.98 \cdot 10^{-3} \cdot 5.53 \cdot 1440 \simeq \simeq 23.7\,\texttt{[min/year]}.$$

Further, the average difference between the downtimes of P_{indep} and P_{edm} in Interoute for *st*-pairs where $P_{\text{indep}} \neq P_{\text{edm}}$ calculated similarly as above turned out to be 14 `[min/year]`, that is still significant. Note that while on the special case depicted in Fig. 4, P_{edm} turned out to be physically longer, the average length of the paths yielded by the EDM is slightly shorter com-

Table 1. Overall description of the real networks, simulations, and path availability comparison

Network	Number of nodes $\|V\|$	Number of edges $\|E\|$	Average internode dist. [km]	Average node degree	Number of sim.	Average CFP	Max. CFP	$A(P_{\text{indep}})$ higher	Avb. equal	$A(P_{\text{edm}})$ higher	$A(P_{\text{edm}})$ higher avg. diff.
Interoute (Italy)	25	34	529	2.72	300	$2.35{\cdot}10^{-3}$	0.025	0	279	21	$1.76 \cdot 10^{-3}$
nfsnet (USA)	79	108	1834	2.73	3081	$3.49{\cdot}10^{-4}$	0.015	0	2040	1041	$8.29 \cdot 10^{-5}$
janos-us (USA)	26	42	1763	3.23	325	$4.96{\cdot}10^{-4}$	0.022	0	305	20	$6 \cdot 10^{-6}$
cost266 (EU)	37	57	1213	3.08	666	$4.74{\cdot}10^{-4}$	0.005	0	633	33	$2 \cdot 10^{-6}$
optic-eu (EU)	22	45	1318	4.09	231	$5.33{\cdot}10^{-5}$	0.005	0	230	1	$2 \cdot 10^{-6}$

pared to those provided by the IM (cf. Fig. 5, and the upcoming paragraph on Path length comparison).

In the other networks, the expected yearly downtime improvement of P_{edm} compared to P_{indep} was more modest, and did not exceed 14 minutes for any *st* point pair. This phenomenon may be attributed to the significantly lower average probability of node destruction by the next earthquake in these networks, as indicated by the average CFP presented in Table 1. Because the effect of our solutions is the most prominent in the Interoute network, in the following, we elaborate on the results exclusively from that.

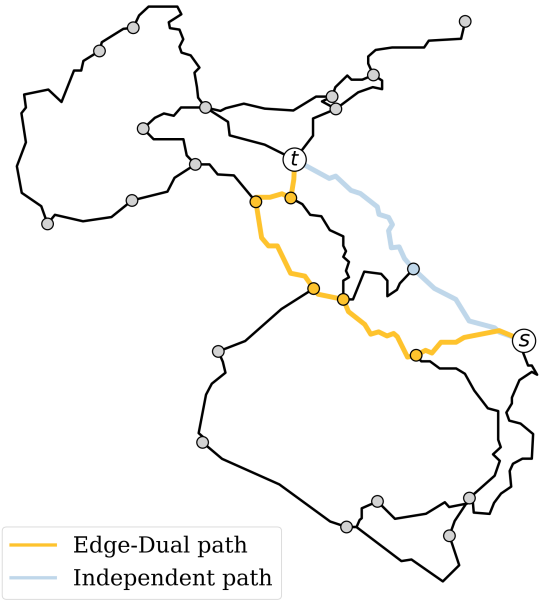


Figure 4. In the example depicted above, the Edge-Dual path yielded by our heuristic has 23.7 minutes less expected yearly downtime due to earthquakes than the Independent path yielded by the traditional approach.

4.4. Path length comparison

In this subsection, the lengths of the paths identified by IM and EDM are compared. In addition, as a baseline, the length of the shortest paths (SPs) is also presented. On Figure 5, the distributions of the length of the paths are visualized where $A(P_{\text{indep}}) \neq A(P_{\text{edm}})$, implying the paths are not the same. The average lengths of the safest paths are 1167 [km], 1193 [km], and 908 [km] for EDM, IM, and SP, respectively. As the EDM and IM visibly choose paths with similar lengths, thus no significant trade-off should be considered between the two methods. Not surprisingly, our methods always recommend paths that have a length equal to or slightly higher than the length of the shortest path.

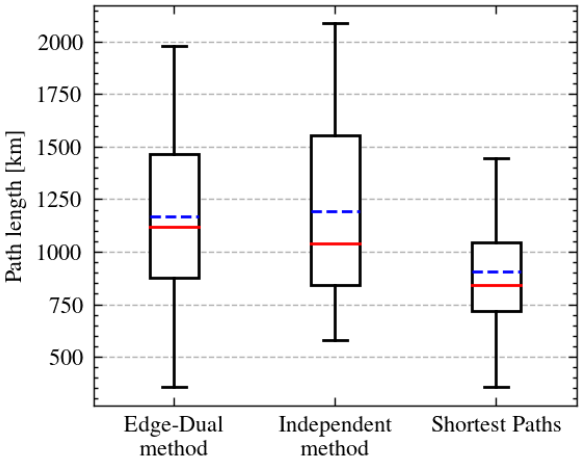


Figure 5. Distribution of the path lengths, exclusively in the simulations where the EDM and the IM returned with different *st*-paths. The median values are indicated by the red lines, and the average values by the blue dashed lines.

4.5. Runtime comparison

To evaluate the runtime performance, we executed all simulations 10 times and measured the elapsed time. In every simulation, both methods were successfully executed under 1 second. However, the EDM involves somewhat more computational steps; thus, the time taken to find the safest path using the IM was consistently shorter than that using the EDM. On average, the IM was faster by factors of 1.6, 5.8, 3.2, 3.0, and 9.8 in the Interoute, nfsnet, janos-us, cost266, and optic-eu networks, respectively. Also supported by the theoretical complexity results, this suggests that apart from a small constant factor, the two compared methods have the same empirical time complexities.

5. Evaluation Results on Synthetic Networks

In this section, firstly we present the methodology of the synthetic data generation, then the performance of the discussed algorithms on procedurally generated network topologies is discussed. These additional results are intends to validate and ex-

tend our findings from real-world networks. The synthetic networks provide controlled environments with varying geometric characteristics and seismic profiles, enabling comprehensive algorithm assessment across diverse network configurations.

5.1. Synthetic data generation

This subsection presents the methodology for the synthetic landmass, network topology, and seismic hazard event generation. The entire process can be divided into 3 times 2 steps, which result in a procedurally generated landmass with a communication network and earthquakes associated to it.

5.1.1. Geography Generation

Landmass Geometry Initialization. To create a landmass with realistic boundaries, we employed a two-step approach that combines gradient functions and Perlin noise [60]. Firstly, a gradient matrix is generated (see Figure 6/A), which contains the gradient values. It is shaped based on one of the three supported landmass shapes: circular, rectangular, or elliptical. The gradient matrix is a smooth gradient with maximum values at the center and minimum values at the domain boundaries. Next, in order to create natural boundary irregularities, a fractal Perlin noise with 5 octaves is generated at the same resolution as the gradient matrix (see Figure 6/B). The number of octaves gives the total number of noise layers combined. After that, the two matrices with the same dimensions are combined multiplicatively to create the final shape of the landmass. Ultimately, a cutoff threshold parameter is employed to regulate the ratio of land to water. An example for a final landmass can be seen on Figure 6/C.

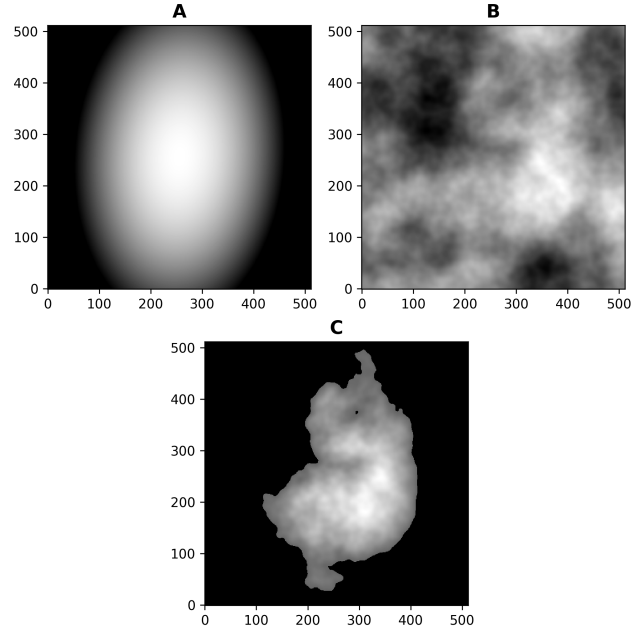


Figure 6. Example visualizations for a gradient (A), a Perlin noise (B) and a landmass matrix (C).

Geometric Scaling and Resolution Enhancement. The algorithm adapts the synthetic geography to realistic geographic scales through intelligent upsampling. Based on predefined country size categories (*small*, *medium*, and *large*, with areas lying in intervals $(10^5, 2\times10^5)$, $(2\times10^5, 10^6)$, and $(10^6, 10^7)$ $[\mathrm{km}^2]$, respectively), the system calculates appropriate zoom factors. Grid unit lengths are derived to ensure proper geometric scaling.

5.1.2. Disaster Data Generation

Seismic Hazard Distribution Modeling. In this step, the algorithm establishes a spatially varying seismic risk profile across the previously created synthetic landmass. To this end, a secondary Perlin noise field is generated and superimposed onto the island geometry to create heterogeneous seismic hazard zones. After normalization and enhancement of high-risk regions, this will be used as a disaster probability matrix. Next, earthquake events are spatially distributed according to the disaster probability matrix, ensuring realistic clustering in high-hazard zones. These seismic events are sampled from an earthquake catalog, which is uniquely generated for each using the Gutenberg-Richter law [61].

Seismic Rate Smoothing. To convert discrete earthquake events into continuous hazard fields suitable for infrastructure planning, the algorithm applies spatial smoothing techniques. The algorithm implements a distance-weighted smoothing kernel according to [10], and it is calculated for each location.

5.1.3. Network Generation

Network Node Placement. Strategic placement of a selected number of network nodes follows an iterative approach that balances coverage requirements with engineering feasibility. At each iteration, the algorithm chooses an available position for the new node using weighted random sampling. The weights are the values of the upscaled version of the Perlin noise matrix, which was used in the *Geography Generation* step (Sec. 5.1.1). Each node placement triggers the creation of a rectangular area with a predefined size, the so-called *no-go zone*, where the placement of additional network nodes is banned. In other words, the no-go zones are rectangular regions around the nodes where no additional nodes are permitted. In addition, a *free range zone* reaching beyond the no-go zone is created around each node, where the new nodes may be placed. An exception is the very first node which can be placed anywhere. As the number of placed nodes is increasing, the size of the free range zone is set to increase too. This intuitively makes more surface available for new nodes. The process ends when the desired number of network nodes is achieved or no new node can be placed due to the no-go zones.

Network Edge Placement. The final network structure is created through a two-phase process combining minimum spanning tree connectivity with strategic edge addition. In the first phase, the initial edges between the nodes are established using a minimum spanning tree construction [62]. This ensures basic connectivity between all nodes with minimal total edge length. In the second phase, the algorithm iteratively adds a maximum selected number of new edges (eg. 10). At each iteration, the edges are ranked based on multiple criteria, and the top one is

selected from the current set of potential edges. Edges that are expected to produce a greater reduction in the average shortest path length are assigned higher rankings. Furthermore, edges that do not increase the nodes' degrees beyond a given number (eg. 4) and that do not introduce crossing edges are also prioritized.

As an example outcome of the procedure, a synthetic network on a generated landmass is shown in Figure 7. This landmass itself was generated using the elliptical gradient and the Perlin noise which is shown in Figure 6. It is a medium sized landmass with an area of 109549 [km^2], on which the resulting network comprises 19 nodes and 22 edges.
Ultimately, the synthetic network and the smoothed seismic rates are processed according to [10] to create the CFP values for the path-finding framework.

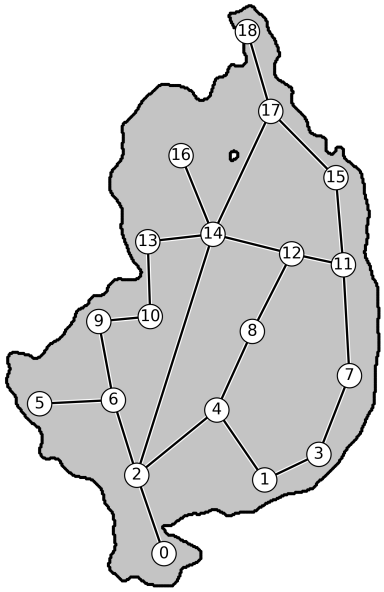


Figure 7. Example for a generated elliptical landmass with the synthetic network.

5.2. Synthetic network settings

Based on the description of the real networks (Table 1), we generated new networks in three groups that would be significant additions to the real networks:

- Networks with smaller average internode distance than 400 [km] (s-network-1/2/3)
- Networks with average internode distance between (700, 1000) [km] (m-network-1/2/3)
- Networks with more than 80 nodes (l-network-1/2/3)

For each group, 3 landmasses and networks were generated with different random seeds. Details about the generated networks are presented in columns 1-5 of Table 2.

The synthetic networks were generated using the methodology described in Section 5.1. Each network incorporates realistic seismic hazard modeling with varying intensity distributions and failure probabilities. Statistics on this hazard data (the average and maximal CFP values) are presented in Table 2. The synthetic networks exhibit different characteristics in terms of connectivity patterns, geographical constraints, and seismic risk profiles. The generated networks range from small-scale topologies with 20 nodes to larger networks with up to 99 nodes, providing diverse testing scenarios.

The results included in Table 2 and discussed in the followings, to demonstrate that the proposed EDM algorithm's advantages are not artifacts of specific real-world network characteristics but represent fundamental algorithmic improvements that generalize across diverse network topologies and geographically correlated seismic risk profiles.

The simulations for the generated networks and the generation processes were conducted on the same virtual machine which is described in Section 4.1.

5.3. Path availability comparison

Table 2 presents a comparison of IM and EDM for the procedurally generated networks based how many cases does one of the methods yields higher availability. The results demonstrate that the pattern favoring EDM observed in real-world networks is also present in most synthetic environments. However, in the **l-network-1**, the IM outperforms EDM in a small fraction of the simulations. In the 23 cases the IM provides higher availability by $4.41 \cdot 10^{-5}$ on average while these paths are on average 1.04 steps longer than the paths recommended by the EDM. Furthermore, it is worth noting that these 23 paths are not independent cases in different parts of the **l-network-1**. All 23 exceptional cases occur at a single face F of **l-network-1**, where the IM and the EDM connect the same two nodes along opposite sides of F. For the path chosen by IM the nodes are 46–50–35–36 and for the EMD are 46–42–36. So each method goes from node 46 to node 36 but in those different ways. Outside F the paths coincide, so the outcome is decided on these two segments. Each method returns the route that its own availability measure rates higher. The discrepancy arises because the EDM estimate A_{ed} is exact ($A(P) = A_{\text{ed}}(P)$) only if Property 1 holds. The EDM's route meets only star-like sets, whereas the IM's route crosses non-star-like sets. Consequently the EDM underestimates the true availability of the IM's route and does not select it, although that route is safer.

In the case of the synthetic networks, the range of the average availability improvements (i.e., the average of $A(P_{\text{edm}}) - A(P_{\text{indep}})$ on the cases where $A(P_{\text{edm}}) > A(P_{\text{indep}})$) is in a similar range as the real-world networks. The biggest availability improvement among the synthetic networks is $1.43 \cdot 10^{-3}$ occurred at the **s-network-2** network, which is comparable to the Interoute network's average improvement of $1.76 \cdot 10^{-3}$. While other networks show smaller differences like the **m-network-1** network with $8.01 \cdot 10^{-5}$.

5.4. Path length comparison:

Analysis of path lengths in synthetic networks reveals patterns similar to those observed in real-world results. The EDM and IM produce paths with comparable lengths, the average

Table 2. Comparison of path availability for synthetic networks

Network	**Number of nodes** $\lvert V\rvert$	**Number of edges** $\lvert E\rvert$	**Average internode dist. [km]**	**Average node degree**	**Number of sim.**	**Average CFP**	**Max. CFP**	$A(P_{\text{indep}})$ **higher**	**Avb. equal**	$A(P_{\text{edm}})$ **higher**	$A(P_{\text{edm}})$ **higher avg. diff.**
s-network-1	22	26	247	2.36	231	$1.40{\cdot}10^{-5}$	0.038	0	199	32	$9.18{\cdot}10^{-4}$
s-network-2	30	32	381	2.13	435	$1.51{\cdot}10^{-4}$	0.035	0	365	70	$1.43{\cdot}10^{-3}$
s-network-3	20	26	252	2.60	190	$2.70{\cdot}10^{-6}$	0.037	0	185	5	$4.85{\cdot}10^{-5}$
m-network-1	50	59	714	2.36	1225	$3.88{\cdot}10^{-4}$	0.012	0	1210	15	$8.01{\cdot}10^{-5}$
m-network-2	50	56	782	2.24	1225	$6.67{\cdot}10^{-4}$	0.011	0	1008	217	$7.37{\cdot}10^{-5}$
m-network-3	45	49	913	2.18	990	$8.81{\cdot}10^{-4}$	0.024	0	891	99	$1.08{\cdot}10^{-4}$
l-network-1	90	111	597	2.47	4005	$4.35{\cdot}10^{-5}$	0.020	23	3335	647	$6.41{\cdot}10^{-4}$
l-network-2	99	119	1073	2.40	4851	$2.69{\cdot}10^{-4}$	0.010	0	4613	238	$2.05{\cdot}10^{-4}$
l-network-3	84	101	733	2.40	3486	$9.41{\cdot}10^{-5}$	0.031	0	2902	584	$6.97{\cdot}10^{-4}$

absolute path length difference across all of the synthetic networks is 0.5km. This confirms that the availability improvements achieved by EDM do not come at the cost of significantly longer routing paths.

5.5. Runtime comparison

Overall, in all of the synthetic network simulations regarding the runtime we observed similar results to the runtime of the real-world networks. All of the simulations for both methods were successfully executed under 1 second and the EDM took more time on average than the IM to find the safest paths. On average, the IM was faster than the EDM by a factor of 1.56.

6. Comparison of Results on Real and Synthetic Data

The synthetic settings provide several key insights that complement and extend the real-world findings:

1. *Algorithm robustness:* The consistent performance across diverse synthetic settings confirms that EDM's superiority is not an artifact of specific topological or seismic features, within the geographically correlated settings considered.

2. *Range validation:* Availability improvements under the investigated synthetic settings lie in a similar order of magnitude as in the real scenarios. In this regard, some synthetic cases are comparable to the best-performing real one (the Interoute network).

3. *Scale independence:* The measure by which the EDM in the great majority of the cases outperforms the traditional IM is independent of the network size.

4. *Geometric independence:* The success across network topologies generated on top of approximately circular, elliptical, and square landmasses, respectively, demonstrates that the algorithm's effectiveness is independent of the underlying geometric constraints.

Table 3. Overall description of the link sets of all CFP[G] from the examined networks.

Network (G)	**Number of all link sets in CFP**[G]	**Number of link sets in CFP**[G] **with Property 1**
Interoute (Italy)	11923	156
nfsnet (USA)	14198	461
janos-us (USA)	750	254
cost266 (EU)	5637	289
optic-eu (EU)	6376	874
s-network-1	66558	84
s-network-2	4663	96
s-network-3	196615	83
m-network-1	942	199
m-network-2	587	186
m-network-3	330	164
l-network-1	71647	413
l-network-2	1393	379
l-network-3	77450	361

The above insights imply the following. The synthetic network results suggest that network designers can expect reliable performance improvements from EDM across the geographically correlated failure scenarios considered, regardless of the specific topological constraints. The controlled synthetic environment demonstrates that the availability improvements observed in real networks represent conservative estimates of the potential benefits, as synthetic networks with more extreme topological features can achieve similar or greater improvements. Furthermore, the consistency of results across different synthetic network sizes indicates that EDM has a scale-independent advantage for both small regional networks and larger continental-scale infrastructures.

Furthermore, it is worth noting that the presented results are based on link sets, which most of the time do not fulfill Property 1. Table 3 shows that the number of the link sets with Property 1

is a fraction of all available sets in all networks. Despite the absence of Property 1 fulfilling sets, the results in Sections 4 and 5 show empirical validation of the efficacy of the EDM and its superiority over the IM in the case of real-world and synthetic networks. Moreover, to examine how the absence of Property 1 affects the relationship between $A(P)$ and $A_{\text{ed}}(P)$, we compared the values of these measures across all simulations. In 28% of the simulations the $A_{\text{ed}}(P)$ underestimated the real availability, $A(P)$. In the remaining 72% of the simulations, the two values were either equal or differed only on the order of 10^{-16}, which is negligible. This surprising effectiveness of the EDM can be attributed to the fact that, in the failure data sets used, larger link sets typically have much lower failure probabilities than their subsets.

All the settings investigated in this paper exhibited some degree of geographical correlation among the failed link sets. We stress that the exactness of the EDM is governed by Property 1, not by the type of hazard. Property 1 captures failures localized at a single network site e.g. joint failures of links meeting at a common node. Geographically regional hazards (earthquakes, and to a large extent floods and hurricanes) tend to produce small-diameter, predominantly star-like failure sets, which is why the EDM stays exact or near-exact there. In contrast, for correlations not tied to geographic proximity (e.g., coordinated malicious attacks, or shared-component/firmware failures spanning arbitrarily distant links), Property 1 is grossly violated. There the EDM keeps its favourable runtime but forfeits its exactness guarantee and is only a heuristic. Since both methods are fast, one can then compute both the EDM and the IM path and return the safer one, guaranteeing performance no worse than the traditional method.

7. Conclusion

This study investigates the problem of finding a safest path between two given network nodes s and t in the case when the failures of network elements are geographically correlated. First, we show that this problem is not only $\mathcal{NP}$-hard, but it does not admit a Fully Polynomial Time Approximation Scheme (FPTAS) either, unless $\mathcal{P} = \mathcal{NP}$.

On the positive side, we investigated the effectiveness of a novel approach (the EDM) for determining the safest paths in real networks in earthquake-hazard areas. Importantly, if the failed network edges are incident to the same node, our method is guaranteed to find a safest st-path. The method uses a simple graph transformation coupled with a cost function computed based on the joint network element failure probabilities that are considered known as part of the problem input. Evaluations show that the EDM is able to find paths that are safer than those determined using a traditional approach that supposes that network element failures are independent. Note that the length of the paths yielded by our and the traditional method are different, but it is not significant. The theoretical worst-case time complexity of our approach is squared in the number $|V|$ of network nodes. Furthermore, under some practical assumptions, this complexity is $O(|V| \log^{2/3} |V|)$. Its execution time on our virtual machine was less than 1 second for every simulation setting, making it a viable alternative to the traditional approach.

One limitation of the present study is that it was designed and tested with a focus on earthquake hazards, whereas other hazards, such as hurricanes or floods, are also can be important factors in finding the safest paths. Moreover, the present results were compared with only one other path-finding method. To further strengthen the findings, comparisons with additional state-of-the-art path-finding methods are necessary.

Further exhaustive comparisons of safest-path-finding algorithms across various hazard datasets can therefore serve as a straightforward direction for follow-up studies.

Data availability

All network and network hazard data used in this paper are accessible at the following GitHub repository: `https://github.com/revpeter/safest_paths_data`.

Acknowledgments

Project no. 153148 have been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the NKKP STARTING_25 funding scheme. This work was supported by a grant of the Romanian Ministry of Research, Innovation and Digitization, CNCS-UEFISCDI, project number PN-IV-P2-2.1-TE-2023-1977.

References

[1] B. Vass, P. Revisnyei, A. Pašić, Computing safest st-paths in backbone networks: Efficiently solvable cases and fast heuristics, in: 2024 14th International Workshop on Resilient Networks Design and Modeling (RNDM), IEEE, 2024, pp. 1–8.

[2] P. Chołda, J. Tapolcai, T. Cinkler, K. Wajda, A. Jajszczyk, Quality of resilience QoR as a network reliability characterization tool, IEEE Network Magazine 23 (2) (2009) 11–19.

[3] Y. Liu, D. Tipper, P. Siripongwutikorn, Approximating optimal spare capacity allocation by successive survivable routing, in: INFOCOM 2001, 2001, pp. 699–708.

[4] V. Y. Liu, D. Tipper, Spare capacity allocation using shared backup path protection for dual link failures, Computer Communications 36 (6) (2013) 666–677.

[5] W. Fawaz, B. Daheb, O. Audouin, M. Du-Pond, G. Pujolle, Service level agreement and provisioning in optical networks, IEEE Communications Magazine 42 (1) (2004) 36–43.

[6] J. Gozdecki, A. Jajszczyk, R. Stankiewicz, Quality of service terminology in ip networks, IEEE communications magazine 41 (3) (2003) 153–159.

[7] G. Iannaccone, C.-n. Chuah, R. Mortier, S. Bhattacharyya, C. Diot, Analysis of link failures in an ip backbone, in: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, ACM, 2002, pp. 237–242.

[8] RFP RMC Rajkot Way, available: `scribd.com/document/528871820/Rfp-Rmc-Rajkot-Way-Part-2`. Accessed: 2024.

[9] J. Tapolcai, B. Vass, Z. Heszberger, J. Biró, D. Hay, F. A. Kuipers, L. Rónyai, A tractable stochastic model of correlated link failures caused by disasters, in: Proc. IEEE INFOCOM, Honolulu, USA, 2018.

[10] A. Valentini, B. Vass, J. Oostenbrink, L. Csák, F. Kuipers, B. Pace, D. Hay, J. Tapolcai, Network resiliency against earthquakes, in: 2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM), 2019, pp. 1–7. doi:10.1109/RNDM48015.2019.8949088.

[11] B. Vass, J. Tapolcai, Z. Heszberger, J. Bíró, D. Hay, F. A. Kuipers, J. Oostenbrink, A. Valentini, L. Rónyai, Probabilistic shared risk link groups modelling correlated resource failures caused by disasters, IEEE Journal on Selected Areas in Communications (JSAC) - issue on Latest Advances in Optical Networks for 5G Communications and Beyond (2021).

[12] D. Zhou, S. Subramaniam, Survivability in optical networks, IEEE network 14 (6) (2000) 16–23.

[13] O. Crochat, J.-Y. Le Boudec, O. Gerstel, Protection interoperability for WDM optical networks, IEEE/ACM Trans. Netw. 8 (3) (2000) 384–395.

[14] C. S. Ou, B. Mukherjee, Survivable Optical WDM Networks, Springer Science & Business Media, 2005.

[15] A. Somani, Survivability and traffic grooming in WDM optical networks, Cambridge University Press, 2006.

[16] G. Aceto, A. Botta, P. Marchetta, V. Persico, A. Pescapé, A comprehensive survey on internet outages, Journal of Network and Computer Applications 113 (2018) 36–63.

[17] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, M. Jaeger, General availability model for multilayer transport networks, in: Design of Reliable Communication Networks (DRCN), Lacco Ameno, Italy, 2005.

[18] S. Yang, S. Trajanovski, F. Kuipers, Availability-based path selection and network vulnerability assessment, Wiley Networks 66 (4) (2015) 306–319.

[19] J. Tapolcai, L. Rónyai, B. Vass, L. Gyimóthi, List of shared risk link groups representing regional failures with limited size, in: IEEE INFOCOM, Atlanta, USA, 2017.

[20] H.-W. Lee, E. Modiano, K. Lee, Diverse routing in networks with probabilistic failures, IEEE/ACM Trans. Netw. 18 (6) (2010) 1895–1907.

[21] J. Liu, J. Zhang, Y. Zhao, C. Ma, H. Yang, W. Li, J. Xin, B. Chen, Differentiated quality-of-protection provisioning with probabilistic SRLG in flexi-grid optical networks, in: OSA Asia Communications and Photonics Conference, 2013, pp. AF2G–8.

[22] J. Oostenbrink, F. Kuipers, Computing the impact of disasters on networks, ACM SIGMETRICS Performance Evaluation Review 45 (2) (2017) 107–110.

[23] S. Neumayer, G. Zussman, R. Cohen, E. Modiano, Assessing the vulnerability of the fiber infrastructure to disasters, IEEE/ACM Trans. Netw. 19 (6) (2011) 1610–1623.

[24] M. T. Gardner, C. Beard, Evaluating geographic vulnerabilities in networks, in: IEEE Int. Communications Quality and Reliability Workshop (CQR), 2011, pp. 1–6.

[25] S. Trajanovski, F. A. Kuipers, A. Ilić, J. Crowcroft, P. Van Mieghem, Finding critical regions and region-disjoint paths in a network, IEEE/ACM Trans. Netw. 23 (3) (2015) 908–921.

[26] X. Long, D. Tipper, T. Gomes, Measuring the survivability of networks to geographic correlated failures, Optical Switching and Networking 14 (2014) 117–133.

[27] X. Wang, X. Jiang, A. Pattavina, S. Lu, Assessing physical network vulnerability under random line-segment failure model, in: IEEE High Performance Switching and Routing (HPSR), 2012, pp. 121–126.

[28] H. Saito, Analysis of geometric disaster evaluation model for physical networks, IEEE/ACM Trans. Netw. 23 (6) (2015) 1777–1789.

[29] H. Saito, Spatial design of physical network robust against earthquakes, J. Lightw. Technol. 33 (2) (2015) 443–458.

[30] P. Sun, Z. He, R. E. Kooij, P. Van Mieghem, Topological approach to measure the recoverability of optical networks, Optical Switching and Networking 41 (2021) 100617. doi:10.1016/j.osn.2021.100617.

[31] F. Dikbiyik, M. Tornatore, B. Mukherjee, Minimizing the risk from disaster failures in optical backbone networks, J. Lightw. Technol. 32 (18) (2014) 3175–3183.

[32] F. Iqbal, F. Kuipers, Spatiotemporal risk-averse routing, in: IEEE INFOCOM Workshop on Cross-Layer Cyber Physical Systems Security (CPSS), 2016.

[33] S. Paira, M. Chatterjee, U. Bhattacharya, On survivable energy-efficient and crosstalk-aware routing, spectrum and core allocation schemes for dynamic multiclass traffic in SDM-EONs, Optical Switching and Networking 42 (2021) 100630. doi:10.1016/j.osn.2021.100630.

[34] S. Das, M. Chatterjee, Energy efficient resource aware protection with rapid failure recovery in cloud-ready elastic optical networks, Optical Switching and Networking 55 (2025) 100793. doi:10.1016/j.osn.2024.100793.

[35] A. K. Sarma, S. K. Deka, N. Sarma, Priority-aware hybrid protection scheme considering traffic class in SEON based on heterogeneous demands, Optical Switching and Networking 61 (2026) 100877. doi:10.1016/j.osn.2026.100877.

[36] M. F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, B. Mukherjee, Design of disaster-resilient optical datacenter networks, J. Lightw. Technol. 30 (16) (2012) 2563–2573.

[37] I. B. B. Harter, D. Schupke, M. Hoffmann, G. Carle, et al., Network virtualization for disaster resilience of cloud services, IEEE Commun. Mag. 52 (12) (2014) 88–95.

[38] B. Mukherjee, M. Habib, F. Dikbiyik, Network adaptability from disaster disruptions and cascading failures, IEEE Commun. Mag. 52 (5) (2014) 230–238.

[39] R. Souza Couto, S. Secci, M. Mitre Campista, K. Costa, L. Maciel, Network design requirements for disaster resilience in IaaS clouds, IEEE Commun. Mag. 52 (10) (2014) 52–58.

[40] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, G. Zussman, The resilience of WDM networks to probabilistic geographical failures, IEEE/ACM Trans. Netw. 21 (5) (2013) 1525–1538.

[41] P. N. Tran, H. Saito, Geographical route design of physical networks using earthquake risk information, IEEE Communications Magazine 54 (7) (2016) 131–137.

[42] H. Honda, H. Saito, Nation-wide disaster avoidance control against heavy rain, IEEE/ACM Transactions on Networking 27 (3) (2019) 1084–1097.

[43] L. Pašić, A. Pašić, F. Mogyorósi, A. Pašić, FRADIR meets availability, in: 2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020, IEEE, 2020, pp. 1–6.

[44] A. Pašić, R. G. ao Silva, F. Mogyorósi, B. Vass, T. Gomes, P. Babarczi, P. Revisnyei, J. Tapolcai, J. Rak, eFRADIR: An Enhanced FRAmework for DIsaster Resilience, IEEE Access 9 (2021) 13125–13148. doi:10.1109/ACCESS.2021.3050923.
URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9319646`

[45] J. L. Gross, J. Yellen, M. Anderson, Graph theory and its applications, Chapman and Hall/CRC, 2018.

[46] B. M. Waxman, Routing of multipoint connections, IEEE journal on selected areas in communications 6 (9) (1988) 1617–1622.

[47] E. W. Zegura, K. L. Calvert, S. Bhattacharjee, GT-ITM: Georgia tech internetwork topology models (software) (1996).

[48] K. L. Calvert, M. B. Doar, E. W. Zegura, Modeling internet topology, IEEE Communications magazine 35 (6) (1997) 160–163.

[49] A. Medina, A. Lakhina, I. Matta, J. Byers, Brite: An approach to universal topology generation, in: MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, IEEE, 2001, pp. 346–353.

[50] M. B. Doar, A better model for generating test networks, in: Proceedings of GLOBECOM'96. 1996 IEEE Global Telecommunications Conference, IEEE, 1996, pp. 86–93.

[51] S. Soltan, G. Zussman, Generation of synthetic spatially embedded power grid networks, in: 2016 IEEE Power and Energy Society General Meeting (PESGM), IEEE, 2016, pp. 1–5.

[52] B. Vass, J. Tapolcai, E. Bérczi-Kovács, Enumerating maximal shared risk link groups of circular disk failures hitting k nodes, IEEE Transactions on Networking (2021).

[53] E. W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271.

[54] Z. Tasnádi, B. Vass, N. Gaskó, Ant colony optimization algorithm for safest path computation in presence of correlated failures in backbone networks, in: ACM Genetic and Evolutionary Computation Conference (GECCO), Málaga, Spain, 2025.

[55] C. A. Tovey, A simplified np-complete satisfiability problem, Discrete applied mathematics 8 (1) (1984) 85–89.

[56] R. M. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department, Springer, pp. 85–103.

[57] R. P. Stanley, Enumerative Combinatorics, 2nd Edition, Vol. 1 of Cambridge Studies in Advanced Mathematics, Cambridge University Press, 2012.

[58] R. Duan, J. Mao, X. Mao, X. Shu, L. Yin, Breaking the sorting barrier for directed single-source shortest paths, in: Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 36–44. doi:10.1145/3717823.3718179.

[59] M. L. Fredman, R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, J. ACM 34 (3) (1987) 596–615. doi:10.1145/28869.28874.
URL https://doi.org/10.1145/28869.28874

[60] K. Perlin, An image synthesizer, ACM Siggraph Computer Graphics 19 (3) (1985) 287–296.

[61] B. Gutenberg, C. F. Richter, Frequency of earthquakes in california, Bulletin of the Seismological society of America 34 (4) (1944) 185–188.

[62] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, Proceedings of the American Mathematical society 7 (1) (1956) 48–50.