Computing Safest *st*-paths in Backbone Networks: Efficiently Solvable Cases and Fast Heuristics

Balázs Vass*†‡, Péter Revisnyei*, Alija Pašić*
*High Speed Networks Laboratory (HSNLab), Department of Telecommunications and Artificial Intelligence (TMIT), Faculty of Electrical Engineering and Informatics (VIK), Budapest University of Technology and Economics (BME), Budapest, Hungary
†Faculty of Mathematics and Computer Science, Babeș-Bolyai University, Cluj Napoca, Romania
‡HUN-REN-BME Information Systems Research Group, Budapest, Hungary
{vb, revisnyei, pasic}@tmit.bme.hu

***Abstract*—For proper evaluation and optimization of the expected availability of a backbone network service, two related very fundamental modeling and algorithmic questions are the following. 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, we show 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). Due to the inconsistent estimations, the supposedly safest paths yielded by these approaches may turn out to be suboptimal. 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.**

***Index Terms*—survivable routing, safest path computation, disaster resilience, network failure modeling, probabilistic shared risk link groups, PSRLG, seismic hazard**

I. Introduction

Many mission-critical services rely on continuous network connections, the quality of these connections being often measured by their Quality of Resilience (QoR) [1]. Examples of such services include telesurgery and stock market operations, both of which demand high reliability and availability. These requirements depend on the underlying network infrastructure, accurate failure modeling, and effective routing schemes (e.g., protection mechanisms). However, networks are typically designed to handle only single link or node failures [2], or dual-failure scenarios [3], which is inadequate for meeting these stringent demands. Therefore, stricter Service Level Agreements (SLAs) are necessary to meet the requirements of these critical communication services, which are increasingly relied upon by both governments and the general public.

An SLA is a formal contract between a service provider and a subscriber that specifies the Quality of Service (QoS) parameters, also known as Service Level Specifications (SLSs), detailing the technical requirements [4], [5]. The primary metrics typically associated with QoS include packet loss, packet delay, guaranteed throughput, and port availability [6]. Additionally, in 2002, the concept of "service availability" was introduced to measure the fraction of time a service is available to customers [6]. A violation of the service availability may lead to a financial penalty for the communication service provider (CSP), e.g., capped at 15% of its Quarterly Payment [7]. Therefore, to meet availability demands and avoid penalties, CSPs must carefully estimate and optimize the availability of their services, with a particular focus on primary paths that primarily deliver information. This involves considering single, independent, and interdependent multiple network element failures (e.g., those caused by natural disasters). To ensure optimal QoR at minimal cost, CSPs must accurately assess and optimize the availability of any source-destination connection.

Decomposed, the two most important ingredients for a thorough availability estimation are the following:

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.

While recently, paper group [8]–[10] made remarkable progress in issue 1), to the best of our knowledge, a thorough study on efficiently computing a safest *st*-connection (issue 2)) remained open. This paper takes some steps towards tackling this latter issue. In our simulations, our proposed fast algorithm for efficiently computing a safest *st*-path consistently outperformed the traditional approach, always providing at least as safe *st*-paths as the traditional approach did.

A. Related Works

Computing availabilities assuming *independent* single element failures is a well-investigated topic [11]–[16]. Also, dealing with correlated failures has a long history in the form of Shared Risk Groups (SRGs) (e.g., [11]–[14], [17], [18]). Here, an SRG typically consists of a few network components (links and nodes) with considerable risk of failing together. An observation is that, in many applications (such as availability estimations), the failure of a node v has the same effect as

the failure of all the links incident to v. In these cases, it is enough to deal with an appropriate list of Shared Risk *Link* Groups (SRLGs), where each SRLG consists only of links. Probabilistic extensions of the SRLG concept were also proposed [8]–[10], [19], [20]. Notably, [10] proposed a straightforward unified terminology related to the Probabilistic SRLGs (PSRLGs) that will be used in this paper.

A natural approach (also taken by this paper) is to take the disaster scenarios as input [21], that have been carefully precomputed, e.g., based on historical hazard data. Unlike this, much of the work on regional failures took a less principled approach, and tackled the disaster modeling more heuristically in their own way to address their given problem in network planning. Some examples are finding the most vulnerable part of the network [22]–[25], estimating the effect on the network of a random disaster [26]–[28], (re)routing of connections to minimize the impact of disasters [29], [30], and resiliency-aware network design [31]–[34].

As of now, by [10], we have efficient methods to compute and store the link failure correlation (instead of limiting the set of disasters to a small number or wrongly assuming link-failure events to be independent [35]–[37]). These methods were already incorporated in complex frameworks for disaster resilience [38], [39]. However, to the best of our knowledge, the issue of efficiently computing a (truly) safest st-path (issue 2)) remained open.

B. Main Contributions

Motivated by the above, our paper delves into the study of the correctness and efficiency of some safest path computation algorithms. The main contributions of the paper are summarized as follows:

- We show that, by not considering joint failure probabilities, the traditional approaches may significantly misguess 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 propose a fast algorithmic approach, that, for a graph $G = (V, E)$, under some assumptions, returns with a (truly) safest st-path accompanied by its exact expected availability in $O(|V| \log |V|)$.
- We provide a proof of concept implementation and simulation based on real-world network topologies and seismic hazard data.
- Through the simulations, we show that, if the disaster data does not obey Property 1 (to be explained), our proposed method proves to be a good heuristic in guessing a safest path, outperforming the traditional approach.

The rest of the paper is organized as follows. Sec. II introduces our formal model and discusses some basic results. In Sec. III, a fast algorithm is described, that under certain circumstances, is guaranteed to compute a safest st-path. Further, our simulation results are presented in Sec. IV. Finally, Sec. V concludes our paper.

II. Model and Basic Algorithms

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 [10], [39], 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 [40]. 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 [10, 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.

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 ≤ 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., $\rho_P := \max\{S \subseteq E \ | \text{CFP}(S) > 0, \text{ and } S \cap P \neq \emptyset\}$). Lastly, the length of a path P is sometimes denoted by k.

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

Property 1: Any set in CFP$[G]$ is connected, and has a diameter at most 2. That is, 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 [9], 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. II-A, then, we discuss exact methods for path availability computation in Sec. II-B.

A. The standard method 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

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

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 $\sum_{e \in P} \log(1 - p(e))$. Luckily, this is equivalent of finding a path minimizing $\sum_{e \in P} -\log(1 - p(e))$. This can be done by e.g., the Dijkstra algorithm [41], since $-\log(1 - p(e)) > 0$ 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 $\{a, b\}$, and $\{c\}$, respectively. Let $\text{CFP}(a) = \text{CFP}(b) = 0.5$, $\text{CFP}(c) = 0.9$, while the rest of the link sets have CFPs of zero (including $\text{CFP}(a, b) = 0$). 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_1. Thus, the standard method would recommend using path P_1. In reality, however, this is a terrible idea, since, with probability 1, either link a or b of path P_1 is not working, thus its real availability $A(P_1) = 1 - \text{CFP}(a) - \text{CFP}(b) + \text{CFP}(a, b) = 0$. I.e., 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 [9] 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.

B. Exact methods for computing the availability of a path

Given a path P, and known the list $\text{CFP}[G]$ of link sets having a positive CFP, Alg. 1 depicts a very straightforward way of computing the availability $A(P)$. Namely, by iterating through the link sets in $\text{CFP}[G]$, and counting their CFP with the right sign, it calculates the following sum:

$$A(P) = \sum_{S \subseteq P} (-1)^{|S|} \text{CFP}(S), \quad (2)$$

Algorithm 1: Computing the availability of a path P in graph G by scanning through the whole list of PSRLGs

Input: Path $P = \{e_1, \ldots, e_k\}$ in graph $G = (V, E)$, cumulative failure probabilities $\text{CFP}[G]$
Output: Availability $A(P)$ of path P

```
1 A := 1        // For counting the availability
  for S ∈ CFP[G] do
      if S ⊆ P then
          A := A + (−1)^|S| CFP(S)
  return A
```

Algorithm 2: Exact method for computing the availability of a path

Input: Path $P = \{e_1, \ldots, e_k\}$ in graph $G = (V, E)$, cumulative failure probabilities $\text{CFP}[G]$
Output: Availability $A(P)$ of path P

```
1 A := 0        // For counting the availability
2 F := ∅        // The set of CFPs factored in
  return FactorInEdgeSet (∅)
  Function FactorInEdgeSet (S):
3     A := A + (−1)^|S| CFP(S)              // CFP(∅) := 1
4     for i ∈ {1, ..., k} do
          if CFP(S ∪ {e_i}) > 0 and {S ∪ {e_i}} ∉ F then
              F := F ∪ {S ∪ {e_i}}
5             FactorInEdgeSet(S ∪ {e_i})
      return A
```

where $|S|$ denotes the cardinality of S. Note that by the inclusion-exclusion principle [42], this sum correctly assesses the availability of path P. Also, it is not a problem that Alg. 1 possibly neglects some sets $S \subseteq P$, that are not in $\text{CFP}[G]$, since their cumulative failure probability is zero by definition.

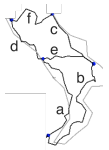
Still, if the cardinality of $\text{CFP}[G]$ is excessive, it can be cumbersome to scan through $\text{CFP}[G]$. To overcome this difficulty, Alg. 2 takes a different approach. It starts exploring the CFPs from the 1-element edge sets. If an edge set S is found that intersects with path P, and it is an element of $\text{CFP}[G]$, it is factored into the sum defined in Eq. 2. Then, the algorithm checks all the link sets that contain an extra link compared to S. If a link set S' is found that is not part of $\text{CFP}[G]$, then Alg. 2 refrains from checking any superset of S', since they will not be part of $\text{CFP}[G]$ anyway.

We can formally state the complexity achieved by the above algorithms as follows.

Theorem 1: The exact availability of a path P can be com-

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

Network G:



$\text{CFP}[G]$:

$\text{CFP}(a) = 4.07 \cdot 10^{-2}$ $\text{CFP}(b) = 3.53 \cdot 10^{-2}$ $\text{CFP}(a, b) = 5.68 \cdot 10^{-3}$ $\text{CFP}(b, e) = 6.91 \cdot 10^{-6}$ $\text{CFP}(a, d, e) = 3.27 \cdot 10^{-4}$ $\text{CFP}(a, b, e) = 0$
$\text{CFP}(c) = 1.13 \cdot 10^{-2}$ $\text{CFP}(d) = 2.91 \cdot 10^{-3}$ $\text{CFP}(a, e) = 4.59 \cdot 10^{-4}$ $\text{CFP}(c, e) = 7.48 \cdot 10^{-4}$ $\text{CFP}(b, c, e) = 6.91 \cdot 10^{-6}$
$\text{CFP}(e) = 1.46 \cdot 10^{-2}$ $\text{CFP}(f) = 2.60 \cdot 10^{-2}$ $\text{CFP}(d, e) = 3.27 \cdot 10^{-4}$ $\text{CFP}(d, f) = 2.78 \cdot 10^{-4}$
$\text{CFP}(c, f) = 5.25 \cdot 10^{-4}$ $\text{CFP}(b, c) = 7.27 \cdot 10^{-6}$
$\text{CFP}(a, d) = 3.35 \cdot 10^{-4}$

Output: a safest st-path

Fig. 1. Example real-world problem input. The depicted network G is a part of the Interoute network [9] covering the southern part of continental Italy. Joint failure probabilities due to earthquakes are also taken from [9], calculated at a shaking intensity tolerance of VI. In this example, Property 1 holds.

puted in $O(|E|\cdot|\text{CFP}[G]|)$, or by denoting the number of edge crossings by x, in $O((|V|+x)\cdot|\text{CFP}[G]|)$. Alternatively, if P has k edges, and by denoting with ρ_P the maximal cardinality of a link set in CFP$[G]$ having a nonempty intersection with P, the availability of P can be computed in $O(\rho_P\cdot(|V|+x)\cdot\binom{k}{\rho_P})$.

Proof: We claim Alg. 1 solves the problem in $O(|E|\cdot|\text{CFP}[G]|)$. Indeed, for each tuple $(S,\text{CFP}(S))$ listed in CFP$[G]$, Alg. 1 has to make $O(|E|)$ operations. Namely, $S\subseteq P$ has to be checked (in $O(|E|)$), and then, possibly, the algorithm makes a constant number of basic arithmetic operations in $O(1)$. We get the $O((|V|+x)\cdot|\text{CFP}[G]|)$ complexity by using the claim that $|E|$ is $O(|V|+x)$ [43, Claim 2].

To prove the alternative complexity results, we have to turn to Alg. 2. Here, starting from the empty set, and adding one new link at a time, the algorithm explores all the sets S that are both subsets of P and have a positive failure probability $\text{CFP}(S)>0$. Supposing that given a link set S, CFP(S) can be looked up in $O(|E|)$, we can see that Alg. 2 has a total complexity of $O(|E|\cdot 2^k)$. This is because the algorithm checks $O(2^k)$ sets, and handling each set takes $O(|E|)$ time. Now, we turn to proving the parametric complexity results.

Trivially, on path P having k edges, there are $\binom{k}{\rho_P}$ link sets having at most ρ_P links. It is easy to see that function `FactorInEdgeSet` is called at most ρ_P times for each set $S\subseteq P$. Each time the function is called for S, it makes $O(|E|)$ operations (the costliest step being looking up CFP(S) in CFP$[G]$).

Thus, Alg. 2 computes the exact availability in $O(\rho_P\cdot|E|\cdot k^{\rho_P})$, or, alternatively, in $O(\rho_P\cdot(|V|+x)\cdot\binom{k}{\rho_P})$. ∎

We note that, as a consequence of the above theorem, if ρ_P is $O(\text{polylog}(|V|))$, Alg. 2 computes the exact availability of a path P in $O(\text{poly}(|V|))$. Also, the complexity results show, that if we, for some practical considerations, would neglect the size of CFP$[G]$ from the problem input, Alg. 2 is still a Fixed Parameter Tractable (FPT) algorithm both in k and ρ_P.

III. A fast algorithm for finding a safest *st*-path

In II-A, 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 computes a safest *st*-path in $O((|V|+x)(\Delta+\log|V|))$ time, that, under practical circumstances, means $O(|V|\log|V|)$.

Our first observation is that, in presence of Property 1, Eq. (2) becomes much simpler:

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

$$A(P)=1-\sum_{j=1}^{i}\text{CFP}(\{e_i\})+\sum_{j=1}^{i-1}\text{CFP}(\{e_j,e_{j+1}\}) \qquad (3)$$

Proof: The proof of Claim 1 relies on the correctness of Eq. (2), supplemented with the following three observations. Firstly, as a consequence of Property 1, only the CFPs of single and double link failures have to be factored in. Secondly, since the diameter of any link set S with positive CFP is at most 2, $S\cap P$ can be either the empty set, a single link, or two consecutive links of path P (with no other options). Thus, in consequence, the CFP of all the link sets that are present in the sums in Eq. (2), but not in Eq. (3) are zero. The proof follows. ∎

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:

$$U(P=\{e_1,\dots,e_i\}):= \\ =\sum_{j=1}^{i}\text{CFP}(\{e_i\})-\sum_{j=1}^{i-1}\text{CFP}(\{e_j,e_{j+1}\}). \qquad (4)$$

Thirdly, instead of minimizing on graph G itself, we will minimize the unavailability in a graph derived from G (see Fig. 2), which is very similar to the so-called line graph of G [44]. 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, $E'_{st}:=E(G)\cup\{\{s,l(e)\}|e=\{s,v\}\in E\}\cup\{\{t,l(e)\}|e=\{t,v\}\in E\}$.

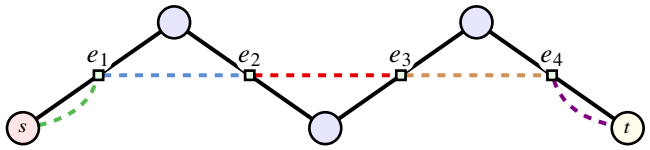
When it does not cause confusion, we refer to the *st*-edge dual simply as the edge dual, and simplify its notation $G'_{st}=(V'_{st},E'_{st})$ to $G'=(V',E')$. Finally, we define a cost function on the edges of G'.

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 $c(\{f',g'\}):=\frac{1}{2}\text{CFP}(f)+\frac{1}{2}\text{CFP}(g)-\text{CFP}(f,g)$. Let the cost of links $\{s',e'\}$ incident to s' be $c(\{s',e'\}):=\frac{1}{2}c(e)$ Similarly, let the cost of links $\{t',e'\}$ incident to s' be $c(\{t',e'\}):=\frac{1}{2}c(e)$.

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 $P=e_1,\dots,e_i$ in G, we define its *counterpart* as $P':=\{s',e'_1\},\{e'_1,e'_2\},\dots,\{e'_{i-1},e'_i\},\{e'_i,t'\}$. 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, $c(P')=U(P)$.



$$U(P) = \text{CFP}(e_1)/2 + \text{CFP}(e_1)/2 + \text{CFP}(e_2)/2 + \text{CFP}(e_2)/2 +$$
$$+\text{CFP}(e_3)/2 + \text{CFP}(e_3)/2 + \text{CFP}(e_4)/2 + \text{CFP}(e_4)/2 -$$
$$-\text{CFP}(\{e_1, e_2\}) - \text{CFP}(\{e_2, e_3\}) - \text{CFP}(\{e_3, e_4\}) = c(P')$$

Fig. 2. 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.

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 $O(|E|^2)$, or $O((|V| + x)(\Delta + \log|V|))$ worst-case time complexity, respectively.

Proof: Due to Cor. 1, it is enough to search for a cheapest st-path in the st-edge dual graph G', since its 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 [41] algorithm to find such a path P'. The best worst case time complexity is $O(m + n\log n)$ [45], where m and n are the number of edges and nodes in the graph, respectively.

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 [18], $|E|$ 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 $O(|E|^2)$, or $O((|V| + x)(\Delta + \log|V|))$ time complexity, respectively. ∎

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 st-regional-dual.

TABLE I
DESCRIPTION OF THE NETWORKS

Network	Number of nodes $\|V\|$	Number of edges $\|E\|$	Number of sim.	Average CFP	Max. CFP
Interoute (Italy)	25	34	300	$2.35*10^{-3}$	0.025
nfsnet (USA)	79	108	3081	$3.49*10^{-4}$	0.015
janos-us (USA)	26	42	325	$4.96*10^{-4}$	0.022
cost266 (EU)	37	57	666	$4.74*10^{-4}$	0.005
optic-eu (EU)	22	45	231	$5.33*10^{-5}$	0.005

IV. Evaluation

In this section, we evaluate the performance of the discussed algorithms. The simulations will be conducted on real-world network topologies, coupled with CFP values distilled from real earthquake disaster datasets (computed in [10], [39]). First, the simulation setting will be described, then, the different availability measures are compared. Next, the performance of the two tackled path-finding methods and the relative lengths of the paths they provide is evaluated, Finally, we disclose the runtimes of the different methods.

Simulation settings: The simulations in this study were conducted using four real-world communication networks. The nfsnet, janos-us, cost266 and optic-eu are obtained from [18], and the Interoute network is taken from [9]. These networks cover the territory of the USA, Europe, and Italy. In this paper, conform to [9], it is expected that network elements fail at a ground movement intensity threshold of VI.

The number of nodes, edges, and simulations for each network are presented in Table I. 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. II-A. 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. III. 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,

(i) $A(P)$ is the (actual) availability defined in Eq. (2),
(ii) $A_{\text{indep}}(P)$ is the availability estimation used IM, defined in Eq. (1), and finally,
(iii) $A_{\text{ed}}(P)$ is the availability estimation used in EDM, defined in Eq. (3).

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)

TABLE II
COMPARISON OF THE AVAILABILITY OF THE *st*-PATHS RETURNED BY OUR HEURISTIC (P_{EDM}) AND THE TRADITIONAL METHOD (P_{INDEP}), RESP.

Network	$A(P_{\text{indep}})$ higher	=	$A(P_{\text{edm}})$ higher
Interoute	0	279	21
nfsnet	0	2040	1041
janos-us	0	305	20
cost266	0	633	33
optic-eu	0	230	1

processor and 16 GB of RAM, running Microsoft Windows 10. The simulation environment and the algorithms are implemented in Python 3.11.5.

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. III, if Property 1 holds, $A_{\text{ed}}(P)$ and $A(P)$ are equal. Our simulations show that even without the fulfillment of Property 1, the differences between these values are in the order of 10^{-16}, which is negligible.

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

Table II 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.

Where $A(P_{\text{edm}})$ is higher, the average availability differences between respective *st*-path pairs are $1.76{\cdot}10^{-3}$, $8.2{\cdot}10^{-5}$, $6 \cdot 10^{-6}$, $2 \cdot 10^{-6}$ and $2 \cdot 10^{-6}$ in the Iteroute, nfsnet, janos-us cost266 and optic-eu networks, respectively.

The difference between the availabilities is the most significant in the case of the Interoute network, 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. 3, 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 $> 4.5 M_w$) [10]. For the sake of estimation, we apply a Mean Time To Repair (MTTR) of 24 hours, equaling 1440 minutes [39] (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:

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

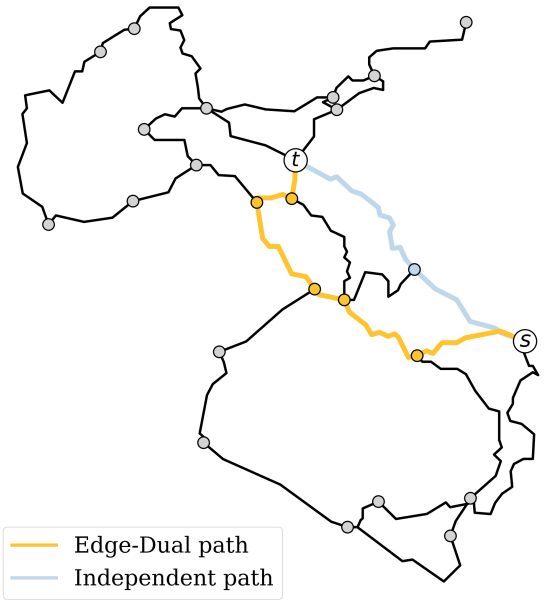


Fig. 3. In the example depicted above, the Edge-Dual path yielded by our heuristic has 23.7 minutes less expected yearly downtime due to earthquakes than the Independent path yielded by the traditional approach.

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. 3, P_{edm} turned out to be physically longer, the average length of the paths yielded by the EDM is slightly shorter compared to those provided by the IM (cf. Fig. 4, and the upcoming paragraph on Path length comparison).

In the other networks, the expected yearly downtime improvement of P_{indep} compared to P_{edm} 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 I. 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.

Path length comparison: In this subsection, the lengths of the paths identified by IM and EMD are compared. In addition, as a baseline, the length of the shortest paths (SPs) is also presented. On Figure 4, 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

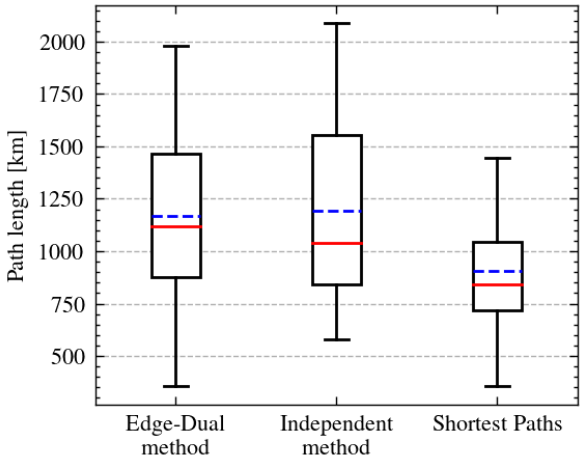


Fig. 4. Distribution of the path lengths, exclusively in the simulations where the EDM and the IM returned with different st-paths. The median values are indicated by the red lines, and the average values by the blue dashed lines.

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.

Runtime comparison: To evaluate the runtime performance, we executed all simulations 10 times and measured the elapsed time. In every simulation, both of the 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 4.4, 4.1, 2.5, 4.8 and 7.2 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.

V. Conclusion and Future Work

In this study, we investigated the effectiveness of a novel approach for determining the safest paths in real networks in earthquake-hazard areas. Our 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. Our evaluations showed that our method, is able to find paths that are safer than those determined using a traditional method. Here, the traditional method (implicitly) supposes that network element failures are independent. Our simulations also showed that, in most cases, this traditional method underestimates the actual availability of the path. Furthermore, our results clearly show that the difference in the length of the paths yielded by our and the traditional method, respectively, is not significant. The execution time of our novel approach, even on a commodity laptop, is less than 1 second for each real-world simulation setting, making it a viable alternative to the traditional approach. A more in-depth empirical comparison of the safest path finding algorithms based on various hazard data sets can be a straightforward aim of follow-up studies.

Acknowledgements

This study has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101155116. Project no. 137698 have been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the PD_21 funding scheme. This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

Co-funded by the European Union

References

[1] P. Chołda, J. Tapolcai, T. Cinkler, K. Wajda, and A. Jajszczyk, "Quality of resilience QoR as a network reliability characterization tool," *IEEE Network Magazine*, vol. 23, no. 2, pp. 11–19, March/April 2009.

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

[3] V. Y. Liu and D. Tipper, "Spare capacity allocation using shared backup path protection for dual link failures," *Computer Communications*, vol. 36, no. 6, pp. 666–677, 2013.

[4] W. Fawaz, B. Daheb, O. Audouin, M. Du-Pond, and G. Pujolle, "Service level agreement and provisioning in optical networks," *IEEE Communications Magazine*, vol. 42, no. 1, pp. 36–43, 2004.

[5] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz, "Quality of service terminology in ip networks," *IEEE communications magazine*, vol. 41, no. 3, pp. 153–159, 2003.

[6] G. Iannaccone, C.-n. Chuah, R. Mortier, S. Bhattacharyya, and 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.

[7] RFP RMC Rajkot Way. Available: scribd.com/document/528871820/Rfp-Rmc-Rajkot-Way-Part-2. Accessed: 2024.

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

[9] A. Valentini, B. Vass, J. Oostenbrink, L. Csák, F. Kuipers, B. Pace, D. Hay, and J. Tapolcai, "Network resiliency against earthquakes," in *2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM)*, Oct 2019, pp. 1–7.

[10] B. Vass, J. Tapolcai, Z. Heszberger, J. Bíró, D. Hay, F. A. Kuipers, J. Oostenbrink, A. Valentini, and 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.

[11] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE network*, vol. 14, no. 6, pp. 16–23, 2000.

[12] O. Crochat, J.-Y. Le Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 384–395, 2000.

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

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

[15] G. Aceto, A. Botta, P. Marchetta, V. Persico, and A. Pescapé, "A comprehensive survey on internet outages," *Journal of Network and Computer Applications*, vol. 113, pp. 36–63, 2018.

[16] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger, "General availability model for multilayer transport networks," in *Design of Reliable Communication Networks (DRCN)*, Lacco Ameno, Italy, Oct. 16-19, 2005.

[17] S. Yang, S. Trajanovski, and F. Kuipers, "Availability-based path selection and network vulnerability assessment," *Wiley Networks*, vol. 66, no. 4, pp. 306–319, 2015.

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

[19] H.-W. Lee, E. Modiano, and K. Lee, "Diverse routing in networks with probabilistic failures," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1895–1907, 2010.

[20] J. Liu, J. Zhang, Y. Zhao, C. Ma, H. Yang, W. Li, J. Xin, and 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.

[21] J. Oostenbrink and F. Kuipers, "Computing the impact of disasters on networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 45, no. 2, pp. 107–110, 2017.

[22] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of the fiber infrastructure to disasters," *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1610–1623, 2011.

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

[24] S. Trajanovski, F. A. Kuipers, A. Ilić, J. Crowcroft, and P. Van Mieghem, "Finding critical regions and region-disjoint paths in a network," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 908–921, 2015.

[25] X. Long, D. Tipper, and T. Gomes, "Measuring the survivability of networks to geographic correlated failures," *Optical Switching and Networking*, vol. 14, pp. 117–133, 2014.

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

[27] H. Saito, "Analysis of geometric disaster evaluation model for physical networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 6, pp. 1777–1789, 2015.

[28] ——, "Spatial design of physical network robust against earthquakes," *J. Lightw. Technol.*, vol. 33, no. 2, pp. 443–458, 2015.

[29] F. Dikbiyik, M. Tornatore, and B. Mukherjee, "Minimizing the risk from disaster failures in optical backbone networks," *J. Lightw. Technol.*, vol. 32, no. 18, pp. 3175–3183, 2014.

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

[31] M. F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee, "Design of disaster-resilient optical datacenter networks," *J. Lightw. Technol.*, vol. 30, no. 16, pp. 2563–2573, 2012.

[32] I. B. B. Harter, D. Schupke, M. Hoffmann, G. Carle *et al.*, "Network virtualization for disaster resilience of cloud services," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 88–95, 2014.

[33] B. Mukherjee, M. Habib, and F. Dikbiyik, "Network adaptability from disaster disruptions and cascading failures," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 230–238, 2014.

[34] R. Souza Couto, S. Secci, M. Mitre Campista, K. Costa, and L. Maciel, "Network design requirements for disaster resilience in IaaS clouds," *IEEE Commun. Mag.*, vol. 52, no. 10, pp. 52–58, 2014.

[35] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "The resilience of WDM networks to probabilistic geographical failures," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1525–1538, 2013.

[36] P. N. Tran and H. Saito, "Geographical route design of physical networks using earthquake risk information," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 131–137, 2016.

[37] H. Honda and H. Saito, "Nation-wide disaster avoidance control against heavy rain," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1084–1097, 2019.

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

[39] A. Pašić, R. G. ao Silva, F. Mogyorósi, B. Vass, T. Gomes, P. Babarczi, P. Revisnyei, J. Tapolcai, and J. Rak, "eFRADIR: An Enhanced FRAmework for DIsaster Resilience," *IEEE Access*, vol. 9, pp. 13 125–13 148, 2021. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9319646

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

[41] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, pp. 269–271, 1959.

[42] Encyclopedia of Mathematics, Inclusion-and-explusion principle. Available at https://encyclopediaofmath.org/index.php?title=Inclusion-and-exclusion_principle, accessed: 2024.

[43] J. Tapolcai, L. Rónyai, B. Vass, and L. Gyimóthi, "Fast Enumeration of Regional Link Failures Caused by Disasters With Limited Size," *IEEE/ACM Transactions on Networking*, vol. 28, no. 6, pp. 2421–2434, 2020.

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

[45] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, p. 596–615, jul 1987. [Online]. Available: https://doi.org/10.1145/28869.28874