This paper was presented as part of the main technical program at IEEE INFOCOM 2011

The Resilience of WDM Networks to Probabilistic Geographical Failures

Pankaj K. Agarwal[†], Alon Efrat[‡], Shashidhara Ganjugunte[†]
David Hay[§], Swaminathan Sankararaman[‡] and Gil Zussman[¶]

[†]Computer Science, Duke University. {pankaj, shashigk}@cs.duke.edu
[‡]Computer Science, University of Arizona. {alon, swami}@cs.arizona.edu
[§] Engineering and Computer Science, Hebrew University. dhay@cs.huji.ac.il*
[¶]Electrical Engineering, Columbia University. gil@ee.columbia.edu

***Abstract*—Telecommunications networks, and in particular optical WDM networks, are vulnerable to large-scale failures of their physical infrastructure, resulting from physical attacks (such as an Electromagnetic Pulse attack) or natural disasters (such as solar flares, earthquakes, and floods). Such events happen at *specific geographical locations* and disrupt specific parts of the network but their *effects are not deterministic*. Therefore, we provide a unified framework to model the network vulnerability when the event has a *probabilistic nature*, defined by an arbitrary probability density function. Our framework captures scenarios with a number of simultaneous attacks, in which network components consist of several dependent subcomponents, and in which either a 1+1 or a 1:1 protection plan is in place. We use computational geometric tools to provide efficient algorithms to identify vulnerable points within the network under various metrics. Then, we obtain numerical results for specific backbone networks, thereby demonstrating the applicability of our algorithms to real-world scenarios. Our novel approach allows for identifying locations which require additional protection efforts (e.g., equipment shielding). Overall, the paper demonstrates that using computational geometric techniques can significantly contribute to our understanding of network resilience.**

***Index Terms*—Network survivability, geographic networks, network protection, computational geometry, optical networks.**

I. Introduction

Telecommunication networks are crucial for the normal operation of all sectors of our society. During a crisis, telecommunication is essential to facilitate the control of physically remote agents, provide connections between emergency response personnel, and eventually enable reconstitution of societal functions. However, telecommunication networks rely heavily on physical infrastructure (such as optical fibers, amplifiers, routers, and switches), making them *vulnerable to physical attacks, such as Electromagnetic Pulse (EMP) attacks, as well as natural disasters, such as solar flares, earthquakes, hurricanes, and floods* [8], [13], [14], [36], [37]. Physical attacks or disasters affect a specific geographical area and will result in failures of neighboring components. Therefore, it is crucial to consider the effect of disasters on the *physical (fiber) layer* as well as on the (logical) network layer.

*This work was done while David Hay was with Columbia University.

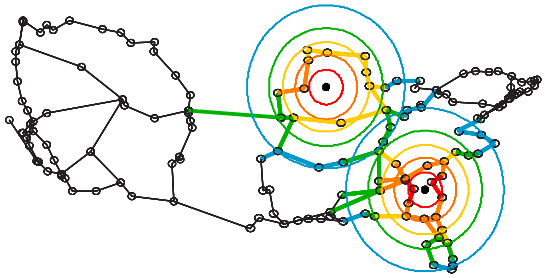

Fig. 1. The fiber backbone operated by a major U.S. network provider [30] and an example of two attacks with probabilistic effects (the link colors represent their failure probabilities).

Although there has been a significant amount of work on network survivability, most previous works consider a small number of isolated failures or focus on shared risk groups (e.g., [6], [22], [27] and references therein). On the other hand, work on large-scale attacks focused mostly on cyber-attacks (viruses and worms). In contrast, we consider events that cause a large number of failures in a specific geographical region.

This emerging field of *geographically correlated failures* has started gaining attention only recently [2], [17], [24]–[26], [31], [32], [37]. However, unlike most of the recent work in this field, we focus on *probabilistic attacks* and on a number of *simultaneous attacks*. Physical attacks rarely have a deterministic nature. The probability that a component is affected by the attacks depends on various factors, such as the distance from the attack's epicenter to the component, the topography of the surrounding area, the component's specifications, and even its location within a building or a system.[1] We consider *arbitrary probability functions* (with constant description complexity) and develop algorithms that obtain the *expected vulnerability* of the network. Furthermore, while [17], [24]–[26], [31], [32], [37] consider only a single event, our algorithms allow the assessment of the effects of *several simultaneous events*.

In particular, we focus on wavelength-routed WDM optical

[1]Characterizing the failure probability function of each component is orthogonal to this research, and we assume it is given as an input.

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.

networks, especially at the backbone [27]. We model the network as a graph, embedded in the plane, in which each node corresponds to an *optical cross-connect (OXC)* and each link corresponds to an *optical fiber* (which are usually hundreds or thousands of kilometers long). Along each link there are *amplifiers*, which are spaced-out approximately equally and are crucial to traffic delivery on the fiber. Data is transmitted on this graph on *lightpaths*, which are circuits between nodes.

While lightpaths can be established by the network dynamically, lightpath-provisioning is a resource-intensive process which is usually slow. If many links fail simultaneously (as in the case of a physical attack or large-scale disaster), current technology will not be able to handle very large-scale re-provisioning (see for example, the CORONET project [10]). Therefore, we assume that lightpaths are *static*, implying that if a lightpath is destroyed, all the data that it carries is lost. Our goal is to identify the most vulnerable location in the network, where vulnerability is measured either by expected number of failed components or by the expected total data loss. In order to do this, our model allows the consideration of failure probabilities of *compound components* by evaluating the effect of the event on their sub-components (e.g., the failure probability of a fiber, due to failure of some amplifiers). Under this model, we develop algorithms that identify the most vulnerable locations with a tradeoff between accuracy and efficiency. Namely, we can provide arbitrarily small errors, albeit high running time. Note that although these algorithms have to be executed offline in preparation for disasters, they have to consider numerous options and topologies, and therefore, efficiency is important. Moreover, our algorithms also work for the deterministic case, with superior performance compared to previous suggestions [25].

For the case of k simultaneous attacks, we are interested in the k-tuple of the most vulnerable locations. In the simultaneous attack case, the problem is hard not only due to its probabilistic nature but also due to the combinatorial hardness of the deterministic problem. Hence, we develop approximation algorithms for various cases.

We also consider networks which are protected, by a *dedicated path protection* plan (either 1+1 or 1:1 plan [27]). Under such plans, every (primary) lightpath has a predefined backup lightpath, on which data can be transmitted, if the primary lightpath fails. These protection plans are pre-computed before a failure event, and therefore, it is reasonable to assume that they can be applied even after a large-scale set of failures. For these networks, we provide approximation algorithms that identify pairs of vulnerable locations that will have a high effect on both the primary and the backup paths.

With the current technology, large-scale dynamic restoration is mostly infeasible. However, this capability will emerge in future optical networks [10]. For future networks with this capability, network resilience can be measured as the maximum post-attack flow. However, we show that computing this measure is in $\#P$ and hence cannot be found in any reasonable time. We discuss options for mitigating this evaluation barrier.

Finally, we provide numerical results that demonstrate the applicability of our algorithms to real backbone networks. Among other things, we show that even when the approximation algorithms only guarantee low accuracy (thereby, having low running time), the obtained results are very close to optimal. This would allow checking various scenarios and settings relatively fast.

The main contributions of this paper are fourfold. First, this is the first paper to present a general probabilistic model for geographically-correlated failures, as well as efficient approximation algorithms for finding the most vulnerable locations in the network. Second, we provide the first set of algorithms that deal with simultaneous attacks. Third, we provide algorithms that take into account pre-computed protection plans. Finally, the paper demonstrates that computational geometric techniques can significantly contribute to our understanding of network resilience.

The rest of the paper is organized as follows. In Section II, we review related work and in Section III, we present the network model and formulate the problem. In Section IV, we develop algorithms for failures centered at a single location and extend them to multiple locations in Section V. We study the effect of protection and restoration plans in Sections VI and VII. We present experimental results in VIII and conclude and discuss future work in Section IX. Due to space constraints, the proofs are omitted and can be found in [3].

II. Related Work

Network survivability and resilience is a well-established research area (e.g., [6], [27] and references therein). However, most of the previous work in this area and, in, particular in the area of physical topology and fiber networks (e.g., [22]) focused on a small number of fiber failures (e.g., simultaneous failures of links sharing a common physical resource, such as a cable, conduit, etc.). Such correlated link failures are often addressed systematically by the concept of *Shared Risk Link Group* (SRLG) [19]. In addition, there exist works which explore dependent failures but do not specifically make use of the causes of dependence (e.g., [21], [34]).

In contrast with these works, we focus on failures within a specific geographical *region* (e.g., [7], [14], [36]), implying that the failed components do not necessarily share the same physical *resource*. To the best of our knowledge, *geographically correlated failures* have been considered only in a few papers and under very specific assumptions [17], [24]–[26], [31], [37]. In most cases, the assumption is that the failures of the components are deterministic and that there is a single failure. Perhaps the closest to the concepts studied in this paper are the problems studied in [7], [25], [32]. In particular, [25] recently obtained results about the resilience of fiber networks to geographically correlated failures where disasters are modeled as circular areas in which the links and nodes are affected. However, [25] considers only a *single disaster* scenario where failures are *deterministic*.

Another closely related theoretical problem is the *network inhibition problem* [28], [29], in which the objective is to minimize the value of a maximum flow in the graph, where

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.

there is a cost associated with destroying each edge, and a fixed budget is given for an orchestrated attack (namely, removing a set of edges whose total destruction cost is less than the budget). However, previous works dealing with this setting and its variants (e.g., [9], [29]) did not study the removal of (geographically) neighboring links.

Notice that when the logical (i.e., IP) topology is considered, wide-spread failures due to attacks by viruses and worms *rather than due to physical attacks* have been extensively studied (e.g., [15]).

III. Problem Statement

Let $G = (V, E)$ be a graph representing the optical network, where V is a finite set of nodes in the plane, and E is a set of links. We assume each link is a straight line segment, and if it is a curve, it can be approximated by a piecewise-linear function. Let $c_e \geq 0$ be the capacity of link e. A lightpath π consists of an ordered sequence of links (where two consecutive links in the path share the same node at one of their endpoints). Let t_π be the amount of data transmitted over π per unit of time. Generally, we consider the nodes and links as *simple components*, and the lightpaths as *compound components*. Let $Q = \{q_1, \ldots, q_m\}$ be a given set of network components. Let w_q be the *weight* associated with a component q, which indicates either the amount of traffic along q or the capacity of the component. Each attack induces a spatial probability distribution on the plane, specifying the damage probability at each location. Taking the perspective of a (simple) component q, given an attack location $p \in \mathbb{R}^2$, let $f(q, p)$ be the probability that q is affected by p. For example, in a deterministic setting, $f(q, p)$ is either 0 or 1. Alternately, one could use more sophisticated models, for example, where $f(q, p)$ depends on the distance from p to q or the length of the portion of link q within the attack radius. In many applications $f(q, p)$ is given, or can be computed as a function of the distance from p to q.

For simple components, we consider only probability functions $f(\cdot, \cdot)$ with *constant description complexity*. Intuitively, these are functions that can be expressed as a constant number of polynomials of constant maximum degree, or simple distributions like the Gaussian distribution. In particular, our results hold for the following important classes of distributions: (i) f is a function of the Euclidean distance, i.e., $f(q, p) = \max\{0, 1 - d(q, p)\}$, where $d(q, p)$ is the Euclidean distance between p and q (more generally, d could be any norm); and (ii) Gaussian distribution, i.e., $f(q, p) = \beta e^{-d(q,p)^2 \alpha}$ for constants $\alpha, \beta > 0$, chosen appropriately to normalize the distribution.

For a compound component (e.g., a lightpath) π, denote by V_π the set of components that constitute the compound component, and for each simple component $v \in V_\pi$, let $f(v, p)$ be the probability that v is affected by p. Thus, the probability that π is affected by an attack in p is

$$f(\pi, p) = 1 - \prod_{v \in V_\pi} (1 - f(v, p)). \quad (1)$$

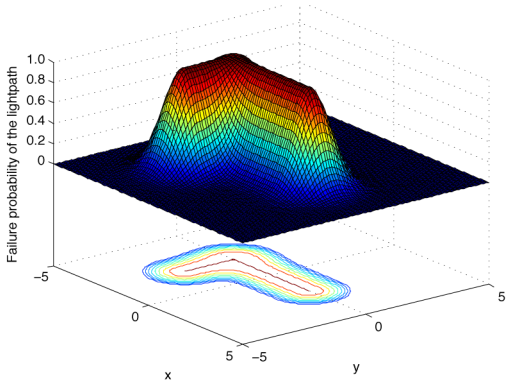

Fig. 2. Function $f(q, \cdot)$, where q is a lightpath that consist on two fibers $[(-2, 0), (2, 0)]$ and $[(-2, 0), (-2, -2)]$ and the attack is Gaussian.

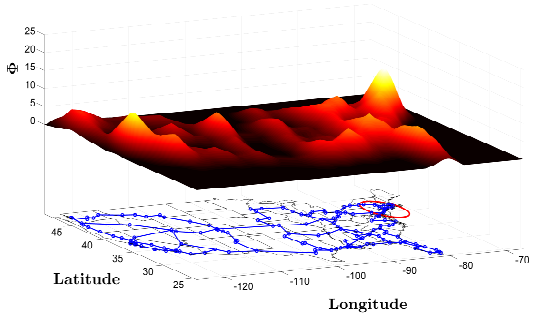

Fig. 3. The function $\Phi(Q, \mathcal{P})$ where the set of components Q are the links that belong to the fiber backbone of a major network service provider [30]. Also shown in red on the network is the location with the highest impact.

See Fig. 2 for an illustration when f is Gaussian. Notice that we can also have a compound hierarchy: for example in certain types of attacks (e.g., EMP attacks), fiber links are not damaged directly. In such cases, we can treat each link as a compound component which consists of several amplifiers (which are simple components represented as points in the plane). Destroying an amplifier along the fiber makes it unusable[2]. Therefore, in such a setting a lightpath is a compound of compound components.

Our goal is to find an attack location (or a set of locations) which has the highest expected impact on the network, where the impact is measured either by the number of failed components (amplifiers, OXCs, or fibers), by the total capacity of failed fibers, or by the total traffic carried by failed lightpaths. For a set of attack locations $\mathcal{P}$, let $\Phi(Q, \mathcal{P})$ denote the expected number (or alternatively, weighted sum) of components of Q affected by the attacks in $\mathcal{P}$ (for an example, see Fig. 3). By linearity of expectation, we get $\Phi(Q, \mathcal{P}) = \sum_{q \in Q} w_q \left(1 - \prod_{p \in \mathcal{P}} (1 - f(q, p))\right)$. When Q is clear from the context, we use $\Phi(\mathcal{P})$. In case a protection plan is in place (see Section VI), we assume that data is lost, if and only if both the primary and backup lightpaths are affected.

The weight w_q of each component enables us to define

[2]More generally, we can treat each link as comprised of a finite set of sample points.

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.

various measures in a unified manner: if Q is the set of amplifiers and w_q is set to 1 (for all q), then $\Phi(Q, \mathcal{P})$ is the expected number of affected amplifiers had the attacks occurred at $\mathcal{P}$. Similarly, if Q is the set of fibers and for any fiber q, $w_q = c_q$ (q's capacity), then $\Phi(Q, \mathcal{P})$ yields the expected capacity loss of attacks in $\mathcal{P}$. Finally, if Q is the set of lightpaths and $w_q = t_q$, then $\Phi(Q, \mathcal{P})$ is the expected loss in traffic, unless there is a protection (or restoration) plan in place. It is important to notice that, by linearity of expectation, $\Phi(Q, \mathcal{P})$ corresponds to the expected value of the measure under consideration, *regardless of any dependency between the various components in* Q. Therefore, even in the extreme situations in which two components share the same physical resource (e.g., lightpaths that share the same fiber, or fibers that share the same conduit), one can evaluate $\Phi(Q, \mathcal{P})$ by considering each component *separately*.

IV. The Single Location Scheme

In this section, we discuss a setting where there is only a single attack. We first describe intuitively how our algorithms find the location with the highest impact on the network. Then, we provide a detailed description of an algorithm that deals only with simple components, discuss its complexity, and prove its correctness. Finally, we will show two extensions: one is geared towards compound components and the other improves the running time by sampling.

A. Outline of the Algorithm

We first assume that $\mathcal{P}$ is a singleton and will denote the location of the (single) attack by p. Fix a network element q. The *super-level-set* of a function $f(q, \cdot)$ with respect to a real value y is the set of points $p \in \mathbb{R}^2$ such that $f(q, p) \geq y$. Given an accuracy parameter $\varepsilon \in (0, 1)$, and a set of components Q, our algorithm first determines a monotonically decreasing vector Y of *level values*. For each level value in Y, and each network element q, the super-level-sets of $f(q, \cdot)$ are constructed. Note that a point can be in multiple super-level-sets. Next, each super-level-set is a geometric region surrounding q. Note that the region corresponding to some level value y contains all super-level-sets of values $y' < y$. Moreover, if f is monotonic with the distance from q, these regions are contiguous in the plane. However, our algorithm does not require this property. Intuitively, the number of level values determines the accuracy and the running time of our algorithm: the more level values we have, the more accurate our results are but our algorithm will require additional time to complete. Fig. 2 depicts an example of super-level-sets induced by a Gaussian attack on a lightpath; each contour defines the edge of a super-level-set.

The next step is to find a point which maximizes Φ (up to some error). In the uniform case (all weights correspond to 1), this is the point that belongs to the largest number of super-level-sets, each might correspond to a different network element (depending on the vector Y, we might count the number of super-level-sets as a weighted sum). In the non-uniform case, each region of component q can conceptually be viewed as w_q coinciding regions. Thus, having essentially the same problem. Finding a point which belongs to the largest number of regions (usually, referred to as a point of maximum *depth*) is a well-studied problem in computational geometry.

Algorithm 1 MaxImpactLocation : Approximation algorithm for computing the optimum location of a single attack.

```
1: point p procedure MaxImpactLocation(Q,W,f,ε)
2:     wm ← max_{q∈Q,p∈R²} w_q f(q,p)
3:     for each q ∈ Q do
4:         w'_q ← w_q/wm
5:     i ← 0, Y ← ∅, Λ ← ∅
6:     while (1−ε)^i ≥ 1/2|Q| do
7:         Y ← Y ∪ {(1−ε)^i}
8:         i++
9:     for each q ∈ Q do
10:        Λ_q ← ∅
11:        for each y ∈ Y do
12:            λ_qy ← {p ∈ R² | f(p,q) ≥ y}
13:            Λ_q ← Λ_q ∪ {λ_qy}
14:        Λ ← Λ ⋃ Λ_q
15:    V ← Compute_Vertices(Λ)
16:    return arg max_{p∈V} Φ(Q,p)
17: end procedure
```

B. Detailed description of the simple component case

We first present an algorithm for simple components, and then describe the modifications required to handle the case of compound components. Our algorithm (see pseudo-code in Algorithm 1) is similar to a recent algorithm by Vigneron [35] and returns a point p such that $\Phi(p) \geq (1-\varepsilon)\Phi(p^*)$, where $p^* = \arg\max_{p \in \mathbb{R}^2} \Phi(p)$ (namely, the optimal attack location), and $0 < \varepsilon < 1$.[3]

The vector Y, which determines how the plane is divided, is defined in an exponentially decreasing manner, such that its i-th element ($i \geq 0$) is $(1-\varepsilon)^i$. The number of elements of Y is the smallest integer s satisfying $(1-\varepsilon)^s < 1/(2m)$, where m is the number of components. Note that $s = O\left(\frac{\log m}{\varepsilon}\right)$.

Using the functions $f(q, p)$, we find for each component q, the value $\max_{p \in \mathbb{R}^2} f(q, p)$. We then scale (linearly) the weights associated with the network components, so that $\max_{q \in Q, p \in \mathbb{R}^2} w_q f(q, p)$ is 1. We call this weight the *normalized weight* of the component q and denote it by w'_q. The rest of our calculations (e.g., Line 16 of Algorithm 1) are done under this normalized weight function.

The next definitions capture formally the essence of levels and how they divide the plane:

Definition 1. *(i) For a value y and a component q, the* super-level-set $\lambda_{qy} = \{p \in \mathbb{R}^2 \mid f(q, p) \geq y\}$ *is the set of all points with $f(q, \cdot)$-value of at least y;*
(ii) For each component q, $\Lambda_q = \{\lambda_{qy} \mid y \in Y\}$ is the set of all component q's super-level-sets*;*
(iii) $\Lambda = \bigcup_q \Lambda_q$ is the set of all super-level-sets*;*

[3] The algorithm of [35] returns a point p' such that $\Phi(p^*) \leq (1+\varepsilon)\Phi(p')$. In addition, our analysis is slightly simpler, and, in certain cases, we show how to achieve a better running time.

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.

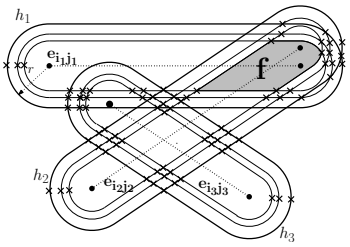

Fig. 4. The arrangement which corresponds to probabilistic attacks of 3 links e_{i_1,j_1}, e_{i_2,j_2}, and e_{i_3,j_3}, such that each has 3 super-level-sets. The shaded region is an example of one of the faces of the arrangement. All vertices of the arrangement are marked with 'x'.

(iv) $\Lambda(p) = \{\lambda_{qy} \in \Lambda \mid p \in \lambda_{qy}\}$ *be the set of all super-level-sets that contain* p*.*

Using the set Λ, whose size is $O(ms)$, we define the *arrangement* $\mathcal{A} = \mathcal{A}(\Lambda)$ of super-level-sets is the subdivision of the plane into vertices, arcs and faces. Under this division: (i) vertices are the set of all intersection points of the boundaries of the super-level-sets along with the set of points where a vertical line is a tangent to the boundary of a super-level-set; (ii) arcs are the maximally connected portions of the boundaries between the vertices; and (iii) faces are maximally connected regions bounded by arcs (for an example, see Fig. 4). Notice that in Algorithm 1 the subroutine **Compute_Vertices**(Λ) returns all the vertices of $\mathcal{A}(\Lambda)$.

Our algorithm computes the arrangement $\mathcal{A}(\Lambda)$ of these $O(ms)$ super-level-sets. The algorithm then evaluates the function $\Phi(\cdot)$ for each vertex of the arrangement, and chooses the best one. The arrangement can be computed in time $O(m \log(m) + \chi)$ [4], [33], where χ is the total number of vertices, arcs, and faces in $\mathcal{A}(\Lambda)$. For simple network components, one can evaluate the function Φ at each vertex naïvely, implying a total complexity of $O(m\chi \log(m))$.

One can improve the running time by refining the faces of the arrangement $\mathcal{A}(\Lambda)$ into cells of constant description complexity (i.e., cells that can described by constant number of polynomial inequalities each of constant maximum degree). This ensures that, in order to compute the function Φ on each cell, there are only a constant number of updates to perform when moving from one cell to another. Thus, by traversing the cells systematically, one can compute the value at each vertex in a constant time, implying a total complexity of $O(m \log m + \chi)$ (see [35] for details). Notice that $\chi = O\left(\frac{m^2}{\varepsilon^2} \log^2 m\right)$, when the super-level-sets themselves have constant description complexity (i.e., as determined by the function f).

Next, we prove the correctness of our algorithm. First, the optimal point is contained within one of the faces of the arrangement $\mathcal{A}(\Lambda)$, because its Φ-value is at least 1 (that is, the maximum normalized weight). On the other hand, one can verify that Φ's value outside the arrangement is at most $1/2$, since the probability to hit any component is less than $1/(2m)$. We prove our approximation ratio by fixing a specific network element and looking at a specific face.

Lemma 1. *If two points* p_1 *and* p_2*, such that* $f(q, p_1) \geq f(q, p_2)$*, are in the same face of* $\mathcal{A}(\Lambda)$*, then, for every component* q*,* $w'_q f(q, p_2) \geq (1 - \varepsilon) w'_q f(q, p_1)$*, where* w'_q *is the normalized weight of* q*.*

Applying Lemma 1 over all the network elements q, we get $\sum_q w'_q f(q, p_2) \geq (1 - \varepsilon) \sum_q w'_q f(q, p_1)$. Since, this approximation is valid for any two points in any face of $\mathcal{A}(\Lambda)$, it also holds for a vertex of the face of $\mathcal{A}(\Lambda)$ and the optimal point. When the function $f(q, p)$ is given by $f(q, p) = \max\{0, 1 - d^2(q, p)\}$, where q is either an amplifier or a link, we can compute the point p^* that maximizes Φ exactly. The key observation is that after partitioning the plane into faces, the gradient inside each face is linear function (see the technical report [3] for details).

C. Extensions to the compound component case

For compound components, one can apply the algorithm MAXIMPACTLOCATION using the function $f(q, \cdot)$ for each compound component q, as defined in (1). Function f takes into account that a single component failure within the compound component is enough to break down the entire component. However, the description complexity of f may not be a constant, thus computing the super-level-sets might be difficult.

To circumvent this problem, we present an algorithm that finds the $(1 - \varepsilon)$-approximated optimal location in two conceptual steps:

1) Compute an arrangement $\mathcal{A}(\Lambda)$ on the $f(v, \cdot)$ for the simple components $v \in V_q$, as defined in Section IV-B. This arrangement deals with the simple component within each compound component directly, both the arrangement size and its computation complexity is the same as in Section IV-B.
2) For every vertex p of a face of $\mathcal{A}(\Lambda)$, we compute the value of $f(q, p)$ using (1). Each such computation requires up to $\kappa = \sum_q |V_q|$ updates to previously obtained value, where κ is the sum of the sizes of all compound components. Thus the total complexity of our algorithm is $O(m \log(m) + \chi\kappa)$.

We now prove the correctness of the above algorithm. Let p be a vertex in the face of the arrangement which contains an optimal point p^*. From Lemma 1, for every simple component v, we have $f(v, p) \geq (1 - \varepsilon) f(v, p^*)$, i.e., $1 - f(v, p) \leq 1 - (1 - \varepsilon) f(v, p^*)$ implying that, for a compound component q, with V_q as components,

$$1 - \prod_{v \in V_q} (1 - f(v, p)) \geq 1 - \prod_{v \in V_q} (1 - (1 - \varepsilon) f(v, p^*)). \quad (2)$$

We also have the following arithmetic lemma:

Lemma 2. *For every point* $p' \in \mathbb{R}^2$*, and a compound component* q *comprised of simple components* V_q*,*

$$1 - \prod_{v \in V_q} (1 - (1 - \varepsilon) f(v, p')) \geq (1 - \varepsilon) f(q, p').$$

Using (2) and applying Lemma 2 for p^*, we get that $f(q, p) \geq (1 - \varepsilon) f(q, p^*)$. The correctness of the algorithm

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.

follows by multiplying by w'_q and summing over all compound components.

D. Improving running time by sampling

Our algorithm MAXIMPACTLOCATION computes all the vertices of the arrangement of super-level-sets, which can be quite large. We present a sampling based algorithm which is significantly faster when the number of vertices that appear on the boundary of the union of super-level-sets Λ is $O(|\Lambda|) = O(m)$, a condition that is true in most practical applications.

Up until now, the errors introduced by our algorithms are a result of *a discretization of function* f, as captured by the different super-level-sets and the corresponding arrangement. Once this discretization is done, we find an *optimal solution with respect to the discrete arrangement.* In this section, we relax the requirement of optimal solution within the arrangement, and require only $(1-\delta)$–approximation. Specifically, we show that by carefully choosing the values of Y and δ, we can obtain a $(1-\varepsilon)$ overall approximation with significantly faster algorithm. Our $(1-\delta)$–approximation of the second stage is conceptually equivalent to finding a point with *approximately maximum depth* within the specific arrangement. This problem was solved by Aronov and Har-Peled [5] through sampling, and therefore, we will build on their technique.

Specifically, fix a level value vector Y, such that the approximation obtained by running the algorithm MAXIMPACTLOCATION is $1-\varepsilon'$. Let y_i denote the i-th value in Y, for $1 \leq i \leq s$. We associate *weights* α_{qy_i} to each super-level-set λ_{qy_i} as follows. We associate to α_{qy_s} the weight $w_q(1-\varepsilon')^{y_s}$, and for $i < s$ we set $\alpha_{qy_i} = w_q(1-\varepsilon')^i - \alpha_{qy_{i+1}}$ Furthermore, for each $p \in \mathbb{R}^2$, let $\Delta_\alpha(p) = \sum_{\lambda_{qy_i} \in \Lambda(p)} \alpha_{qy_i}$. We call this value the *weighted depth* of p with respect to weights α. By the choice of α_{qy_i}, we get $\Delta_\alpha(p) \geq (1-\varepsilon')\Phi(p)$. Thus, the problem of finding a point that approximately maximizes $\Phi(\cdot)$ reduces to finding a point of maximum weighted depth in $\mathcal{A}(\Lambda)$.

Let $\alpha_{\max} = \max_{q \in Q, y_i \in Y} \alpha_{qy_i}$. We scale the weights of the super-level-sets so that the new weights are, $\zeta_{qy_i} = \frac{\alpha_{qy_i}}{\alpha_{\max}} \frac{|\Lambda|}{\varepsilon'}$. This ensures that $\Delta_\zeta(p^*) \geq |\Lambda|/\varepsilon'$, where p^* is a point that maximizes $\Phi(\cdot)$. We round the weights by setting $\beta_{qy_i} = \lfloor \zeta_{qy_i} \rfloor$. Therefore, for any point p, whose depth with respect to weights ζ is at least $|\Lambda|/\varepsilon'$, we have $\Delta_\beta(p) \geq \Delta_\zeta(p) - \varepsilon'|\Lambda|/\varepsilon' \geq \Delta_\zeta(p) - \varepsilon'\Delta_\zeta(p) = (1-\varepsilon')\Delta_\zeta(p)$. Thus, we can assume that the weights of the super-level-sets are integers between 0 and $\lfloor |\Lambda|/\varepsilon' \rfloor$. Let Λ^c be the multiset of super-level-sets obtained by making β_{qy_i} copies for each super-level-set λ_{qy_i}. The (unweighted) *depth* of a point p in $\mathcal{A}(\Lambda^c)$, which we call $\Delta(p)$, is the number of copies that contain p. We can now use the algorithm of Aronov and Har-Peled [5], which works with unweighted depth, to compute a point p such that $\Delta(p) \geq (1-\delta)\Delta(p^*)$. This implies, $\Delta_\zeta(p) \geq \Delta(p) \geq (1-\delta)\Delta(p^*) \geq (1-\delta)(1-\varepsilon')\Delta_\zeta(p^*)$, and after rescaling we get $\Delta_\alpha(p) \geq (1-\delta)(1-\varepsilon')\Delta_\alpha(p^*) \geq (1-\delta)(1-\varepsilon')^2\Phi(p^*)$. We choose $\delta = \varepsilon' = \varepsilon/8$, to get the desired $(1-\varepsilon)$-approximation. Note that if the copies are stored explicitly, we would need $\Omega(|\Lambda^2|/\varepsilon)$ copies in the worst case. Therefore, we show that the copies can be maintained implicitly to achieve a faster expected running time. Namely, near-linear in $|\Lambda|$.

Conceptually, the algorithm of Aronov and Har-Peled works by first generating a random sample $R^c \subseteq \Lambda^c$, i.e., by choosing each copy of a super-level-set in Λ^c with probability ρ. A decision procedure is then invoked to check if the maximum (unweighted) depth in the arrangement of $\mathcal{A}(R^c)$ is at least a threshold $\tau = O(\varepsilon^{-2} \log \frac{|\Lambda|}{\varepsilon})$. If the depth is less than the threshold, ρ is doubled and we repeat this process until either the depth is more than τ or the entire arrangement $\mathcal{A}(\Lambda^c)$ is computed, in which case, a point of maximum depth is returned. Thus, the number of iterations is $O(\log \frac{|\Lambda|}{\varepsilon})$, and the total running time is $O(T_D \log \frac{|\Lambda|}{\varepsilon})$, where T_D is the expected time to choose the random sample and execute the decision procedure to see if the maximum depth is at least τ. We now show that one can maintain the copies implicitly, and execute the decision procedure so that, $T_D = O(\frac{|\Lambda|}{\varepsilon^2} \log^2 \frac{|\Lambda|}{\varepsilon})$. We use the fact that the number of copies of any super-level-set λ_{qy_i} chosen in the random sample R^c follows a *binomial distribution*, $B(\beta_{qy_i}, \rho)$, with parameters β_{qy_i} and ρ. So, for each super-level-set λ_{qy_i} we generate a binomial random variate $\nu_{qy_i} \sim B(\beta_{qy_i}, \rho)$ in $O(\log \beta_{qy_i})$ expected time [12], and associate ν_{qy_i}, as the weight of the super-level-set λ_{qy_i}. Repeating this for each super-level-set, we can generate a set $R = \{\lambda_{qy_i} \mid \nu_{qy_i} > 0\}$, of distinct super-level-sets in expected time $O(|\Lambda| \log \frac{|\Lambda|}{\varepsilon})$. We then use a *randomized divide-and-conquer* algorithm to check if the maximum weighted depth (with respect to weights ν) in the arrangement $\mathcal{A}(R)$ is at most τ. For example, the algorithm of Agarwal et al. [1] can be adopted for this purpose. Since the number of distinct super-level-sets in R is at most $|\Lambda|$, the expected running time of this procedure is $O(\frac{|\Lambda|}{\varepsilon^2} \log^2 \frac{|\Lambda|}{\varepsilon})$. Thus, the overall running time to determine a point p, such that $\Phi(p) \geq (1-\varepsilon)\Phi(p^*)$, is $O(\frac{|\Lambda|}{\varepsilon^2} \log^3 \frac{|\Lambda|}{\varepsilon})$.

V. SIMULTANEOUS ATTACKS

We now consider scenarios in which k attacks may happen simultaneously. Our goal is therefore to identify the set $\mathcal{P}$ of k locations, for which $\Phi(Q, \mathcal{P})$ is maximized over all possible choices of k locations. In general, finding this set $\mathcal{P}$ is NP-hard, since maximizing the value of Φ is a generalization of the well-known *maximum set cover problem* [18]. Nevertheless, we show that the function Φ satisfies certain interesting properties (namely, monotonicity and submodularity). We then present a fast approximation algorithm with a constant approximation ratio, relying on these properties.

A. Algorithm definition

In this section, we formally define the *ε-greedy algorithm* that selects the locations one by one, so as to maximize the gain in Φ (given all past selections).

Specifically, the algorithm works iteratively. Let $\mathcal{P}_{k'} = \{p_1, \ldots, p_{k'}\}$ be the set of locations that were chosen in the first k' iterations. Let $p^* \notin \mathcal{P}_{k'}$ be the location that maximizes

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.

$\varphi(p, \mathcal{P}_{k'}) = \Phi(Q, \mathcal{P}_{k'} \cup \{p\}) - \Phi(Q, \mathcal{P}_{k'})$ over all points $p \in \mathbb{R}^2$. Namely, the location that maximized the *revenue* in terms of Φ. The ε-greedy algorithm chooses at iteration $k'+1$, a location p such that $\varphi(p, \mathcal{P}_{k'}) \geq (1-\varepsilon)\varphi(p^*, \mathcal{P}_{k'})$.

Notice that $\varphi(p, \mathcal{P}_{k'}) = \sum_{q \in Q} \mu(q, \mathcal{P}_{k'}) f(q, p)$, where $\mu(q, \mathcal{P}_{k'}) = w_q \prod_{p' \in \mathcal{P}_{k'}} (1 - f(q, p'))$. This implies that finding the location that maximizes $\varphi(p, \mathcal{P}_{k'})$ within factor $(1-\varepsilon)$ can be done by applying the algorithms of Section IV after modifying the weights of the components to $\mu(q, \mathcal{P}_{k'})$ (instead of w_q).

B. Performance evaluation

As mentioned previously, computing Φ exactly is NP-hard. However, the function $\Phi(Q, \cdot)$ has two key properties, *monotonicity* and *submodularity*, which are used to develop an approximation algorithm. Intuitively, the expected number of failures only increases with the number of attacks. Hence, $\Phi(Q, \cdot)$ is monotonically non-decreasing, i.e., $\Phi(Q, \mathcal{P}_1) \leq \Phi(Q, \mathcal{P}_2)$, for any set $\mathcal{P}_2 \supseteq \mathcal{P}_1$ (formally, this property stems from the fact that $\mu(q, \mathcal{P}_2) \leq \mu(q, \mathcal{P}_1)$, for any $q \in Q$). The function $\Phi(Q, \cdot)$ also exhibits the "law of diminishing returns" property or *submodularity*: for a given attack p and two sets of attacks $\mathcal{P}_1$ and $\mathcal{P}_2$ such that $\mathcal{P}_2 \supseteq \mathcal{P}_1$, the incremental gain of p is lower if it happens after $\mathcal{P}_2$ than if it happens after $\mathcal{P}_1$. The following lemma captures this property.

Lemma 3. *$\Phi(Q, \cdot)$ is a submodular function. Namely, for any two set of points $\mathcal{P}_1$ and $\mathcal{P}_2$, such that $\mathcal{P}_2 \supseteq \mathcal{P}_1$, and any point $p \in \mathbb{R}^2$, $\Phi(Q, \mathcal{P}_1 \cup \{p\}) - \Phi(Q, \mathcal{P}_1) \geq \Phi(Q, \mathcal{P}_2 \cup \{p\}) - \Phi(Q, \mathcal{P}_2)$, i.e., $\varphi(p, \mathcal{P}_1) \geq \varphi(p, \mathcal{P}_2)$.*

These two properties immediately imply that a perfect greedy algorithm (that is, ε-greedy algorithm with $\varepsilon = 0$) achieves a $(1 - 1/e)$-approximation [23]. Since our selection at each step is approximate, the overall approximation ratio of ε-greedy is $(1 - \frac{1}{e^{1-\varepsilon}})$ [16], for any $0 < \varepsilon < 1$. Note that our proof holds for both types of components.

VI. Networks with a Protection Plan

When building a resilient network, a common paradigm is to provide a *protection plan* for significant lightpaths to ensure their continuous service in the face of network failures. Common approaches include 1+1 dedicated protection where, conceptually, the data is sent twice along primary and backup lightpaths, implying that data is lost only when both lightpaths fail simultaneously. A 1:1 dedicated protection, on the other hand, allows using a backup lightpath for low-priority traffic. Once the primary lightpath fails, traffic is shifted to the backup lightpath, and the low-priority traffic is disregarded.

When designing a protection plan, geographical correlation is often taken into account. The primary and backup lightpaths tend to be fiber-disjoint or even to be part of different Shared Risk Link Groups (SRLGs). For example, the fibers should not be close physically. Thus, it is likely that a reasonable protection plan will cope with a single attack. In this section, we are evaluating the resilience of a protection plan to *two simultaneous attacks*.

Formally, we are given pairs of lightpaths (π_i, π'_i), where π_i is the primary path and π'_i is the backup path. Let T_i and t_i be, respectively, the high-priority and low-priority traffic on these lightpaths (for 1+1 protection, t_i is always 0). Thus, one loses t_i when either π_i or π'_i fails, and $T_i + t_i$ if both fail at once. Hence, given two locations p_1 and p_2, the *expected loss on the i-th pair* (obtained by case analysis) is

$$\begin{aligned}\Phi_i(p_1, p_2) = \quad & t_i[1 - g(\pi_i, p_1)g(\pi_i, p_2)g(\pi'_i, p_1)g(\pi'_i, p_2)] \\ & + T_i[f(\pi_i, p_1)f(\pi'_i, p_1) + f(\pi_i, p_2)f(\pi'_i, p_2) \\ & - f(\pi_i, p_1)f(\pi'_i, p_1)f(\pi_i, p_2)f(\pi'_i, p_2) \\ & + g(\pi_i, p_1)f(\pi'_i, p_1)f(\pi_i, p_2)g(\pi'_i, p_2) \\ & + f(\pi_i, p_1)g(\pi'_i, p_1)g(\pi_i, p_2)f(\pi'_i, p_2)] \,, \end{aligned} \tag{3}$$

where $g(\pi, p)$ denotes $1 - f(\pi, p)$. For the entire network, we get $\Phi(p_1, p_2) = \sum_i \Phi_i(p_1, p_2)$. We next show how to compute the pair of attacks p_1, p_2 that maximizes $\Phi(p_1, p_2)$. Notice that one can also measure the *worst-case vulnerability* of the protection plan by the value of $\Phi(p_1, p_2)$ and use this value to compare the resilience of alternative plans.

The algorithm is a generalization of MaxImpactLocation with the following modification: Instead of computing the value of Φ for each vertex of a face of the arrangement, we consider all possible pairs of vertices and compute the value of Φ as though the attacks happened in these locations. This implies that the running time is quadratic in the size of the arrangement. Moreover, the approximation ratio can degrade up to a factor of $(1-\varepsilon)^4$, as we multiply four terms in (3). This can be solved by a refined arrangement defined with $\varepsilon' = 1 - \sqrt[4]{1-\varepsilon} = \Theta(\varepsilon)$, with no extra complexity penalty.

VII. Networks with Restoration Algorithms

An alternative approach to network survivability is to devise a *dynamic restoration scheme*, which, upon a network failure, is able to re-route the traffic so as to avoid data loss. In general, devising efficient restoration algorithms, especially when required to handle large-scale failures, is a challenging task. Dynamic restoration schemes are more efficient in utilizing network capacity, but have slower recovery time and often cannot guarantee quality of restoration.

Clearly, the optimal quality of restoration (in terms of post-attack traffic carried by the network between predetermined source nodes and target nodes) is the *maximum flow* of the residual network and, therefore, finding the most vulnerable location in such setting is equivalent to finding the location whose corresponding attack minimizes the *expected maximum flow*. However, under a probabilistic setting, finding the expected maximum flow of a graph is $\#P$-complete. This is true even if all edges have unit weight (that is, a connectivity problem), and even if the graphs are planar. It is important to notice that although one is not directly required to compute the exact *value* of the expected maximum flow in order to find the most vulnerable location, and, in some cases, one can compare the effects of two locations without such computation

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.

(e.g., when the failure probability of one location dominates the other), in the general case, such computation is necessary (e.g., two locations affecting disjoint sets of links and there is no third location that can be used for comparison). Thus, we obtain the following result.

Theorem 1. *Computing the most vulnerable location in term of expected maximum flow is #P-complete.*

Essentially, this hardness result implies that finding the most vulnerable location requires an exponential-time algorithm *in the number of affected links*. Such algorithms might be feasible to implement when the number of these links is bounded by a small constant s. The most intuitive approach is by a *complete state enumeration*. Such an algorithm considers one candidate location at a time (obtained by the corresponding arrangement, as in Section IV); each location defines a probabilistic graph $G = (V, E)$ where every edge $e \in E$ has a failure probability $\Pr_e$. Let E_1 denote the edges with zero failure probability, and E_2 the rest of the edges. The algorithm will enumerate all subsets of E_2 and for each such subset S, compute the probability for such a failure pattern $\Pr_S = \prod_{e \in S} \Pr_e \prod_{e \in E_2 \setminus S} (1 - \Pr_e)$. Then, it computes the maximum flow F_S in $G_S = (V, E_1 \cup S)$. The expected maximum flow is $\sum_{S \subseteq E_2} \Pr_S \cdot F_S$, and its computation requires $2^{|E_2|} \leq 2^s$ maximum-flow computations.[4]

Alternative techniques, such as graph simplification, graph factoring, and inclusion-exclusion based approaches were also studied in the past [11]. However, all the suggested algorithms still require exponential running time.

VIII. Numerical Results

We have run the algorithms of Section IV on three different networks within the continental USA: Level 3's network of 230 links [20], Qwest's fiber-optic network of 181 links [30], and XO Communications' long-haul network of 71 links [38]. We obtained lightpath information for the last two. In addition, for Qwest's network, we have the transmission rates of individual lightpaths, which were used to determine the values of t_π.

We conducted simulations with five different accuracy values ε for simple components: $0.1, 0.2, \ldots, 0.5$. For compound components, we used three values of ε' (recall Section IV-C): 0.1, 0.2 and 0.3; roughly 0.8-, 0.65-, and $0.5-$approximations. In addition, we considered five different attack radii, ranging between 60 and 300 miles. Finally, two f functions were used: a function that decreases *linearly* with the distance, and a function that follows a *Gaussian* distribution (see Section III).

We first compared the values of Φ for different accuracy values ε of our algorithms. Table I shows the results for simple and compound components when the attack radius (resp., standard deviation of radius) is 180 miles for the linear (resp., gaussian) f-function. Here, Φ_L and Φ_G respectively denote Φ under linear and Gaussian probability functions. Our results show *no perceptible change in* Φ *when* ε *is changed, neither for links nor for lightpaths*. This conclusion holds for all three networks, for both f-functions and for various attack radii. This may be explained by the fact that, in these networks, the location found by MaxImpactLocation lies on, or extremely close to a fiber link, thus avoiding the worst-case (in terms of approximation ratio). While cases where Φ varies significantly with ε do exist, our results show that, *in practice, the dependence on* ε *is very limited*. This implies that for realistic fiber-optic networks, the much faster 0.5-approximation algorithms obtain very close to optimal results.

[4]Note that the arrangement of Section IV induces only an approximate solution. In this case, we need to scale the error parameter ε inversely with s to avoid accumulating errors in the computation.

TABLE I
Values of Φ for simple and compound components under linear f-function (Φ_L) and Gaussian f-function (Φ_G).

	Level3		Qwest		XO	
	Φ_L	Φ_G	Φ_L	Φ_G	Φ_L	Φ_G
Simple comp.	20.5	69.4	14.1	37.2	6.1	15.6
Compound comp.	-	-	475.7	615.1	11.1	15.8

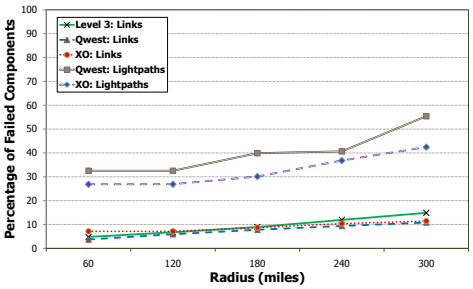

Fig. 5. Variation of Φ, normalized by the sum over the entire network, with the attack radius for a *linear* failure probability function.

To validate our algorithm, we also computed Φ for all three networks when attack locations are restricted to a fine grid of cell size 0.6×0.6 miles. Fig. 3 shows the effects on Qwest's network, of attacks of radius 180 miles centered at locations on this grid. The point corresponding to the maximum value of Φ lies less than 0.5 miles from MaxImpactLocation's output (shown in red in Fig. 3) and the values of Φ are also almost the same. These results further reinforce the conclusion that MaxImpactLocation is, in practice, very close to optimal.

Finally, Fig. 5 shows the change in Φ with the attack radius for a linear f-function for both simple and compound components. We normalized the value of Φ, so that 100% implies the sum of the weights of all network components. As can be seen, the marginal gain in increasing the attack radius is limited, and even small attacks with radius of 60 miles can cause large damage if they are placed in vulnerable locations.

IX. Conclusions

In this paper, we provided a unified framework to identify vulnerable point(s), given a WDM network embedded in the Euclidean plane. A unique feature of our framework is its ability to cope with a *wide range of probabilistic attack and failure models*.

The basic building block of our framework is the algorithm MaxImpactLocation, which locates efficiently a point in the plane that causes arbitrarily close to maximum

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.

impact on a network comprised of simple components. By its tolerance factor ε, MAXIMPACTLOCATION trades accuracy with running time. We further extended and improved MAXIMPACTLOCATION in various ways that allow it to deal with compound components, simultaneous attacks, networks equipped with a protection plan and to deal faster with simpler networks or probabilities. We also evaluated its performance by simulation on three real WDM networks. Our numerical results show, quite surprisingly, that MAXIMPACTLOCATION finds a location very close to optimal, even when taking a high tolerance factor ε (e.g., when it runs very fast but with a loose guarantee on the quality of its output). This makes MAXIMPACTLOCATION an even more attractive tool for assessing network resilience.

Future research directions include developing efficient planning methods for geographically-resilient networks and investigating the effect of adding minimal infrastructure (e.g., lighting-up dark fibers) on network resilience. Finally, we plan to determine how to use low-cost shielding for existing components to mitigate large-scale physical attacks.

ACKNOWLEDGMENTS

The work of P.A. and S.G. is supported by NSF under grants CNS-05-40347, CCF-06-35000, IIS-07-13498, and CCF-09-40671, by ARO grants W911NF-07-1-0376 and W911NF-08-1-0452, by an NIH grant 1P50-GM-08183-01, by a DOE grant OEG-P200A070505, and by a grant from the U.S. Israel Binational Science Foundation. The work of A.E. and S.S. is supported by NSF CAREER grant 0348000 and NSF grant CNS-1017714. The work of G.Z. and D.H. is supported by DTRA grant HDTRA1-09-1-0057, NSF grant CNS- 1018379, CIAN NSF ERC under grant EEC-0812072, and the Legacy Heritage Fund program of the Israel Science Foundation (Grant No. 1816/10). The authors thank the anonymous referees for their valuable suggestions and comments.

REFERENCES

[1] P. K. Agarwal, D. Z. Chen, S. K. Ganjugunte, E. Misiołek, M. Sharir, and K. Tang, "Stabbing convex polygons with a segment or a polygon," in *Proc. ESA*, Sep. 2008.

[2] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "Network vulnerability to single, multiple, and probabilistic physical attacks," in *Proc. MILCOM*, Nov. 2010.

[3] ——, "The resilience of WDM networks to probabilistic geographical failures," Columbia University, Electrical Engineering, Tech. Rep. 2010-07-25, July 2010. [Online]. Available: http://www.cs.huji.ac.il/~dhay/publications/CU-EE-2010-07-25.pdf

[4] N. M. Amato, M. T. Goodrich, and E. A. Ramos, "Computing the arrangement of curve segments: divide-and-conquer algorithms via sampling," in *Proc. ACM-SIAM SODA*, 2000.

[5] B. Aronov and S. Har-Peled, "On approximating the depth and related problems," in *Proc. ACM-SIAM SODA*, Jan. 2005.

[6] R. Bhandari, *Survivable networks: algorithms for diverse routing*. Kluwer, 1999.

[7] D. Bienstock, "Some generalized max-flow min-cut problems in the plane," *Math. Oper. Res.*, vol. 16, no. 2, pp. 310–333, 1991.

[8] J. Borland, "Analyzing the Internet collapse," *MIT Technology Review*, Feb. 2008. [Online]. Available: http://www.technologyreview.com/Infotech/20152/?a=f

[9] R. L. Church, M. P. Scaparra, and R. S. Middleton, "Identifying critical infrastructure: the median and covering facility interdiction problems," *Ann. Assoc. Amer. Geographers*, vol. 94, no. 3, pp. 491–502, 2004.

[10] G. Clapp, R. Doverspike, R. Skoog, J. Strand, and A. V. Lehmen, "Lessons learned from CORONET," in *OSA OFC*, Mar. 2010.

[11] C. J. Colbourn, *The Combinatorics of Network Reliability*. Oxford University Press, 1987.

[12] L. Devroye, *Non-Uniform Random Variate Generation*. Springer-Verlag, 1968.

[13] W. R. Forstchen, *One Second After*. Tom Doherty Associates, 2009.

[14] J. S. Foster, E. Gjelde, W. R. Graham, R. J. Hermann, H. M. Kluepfel, R. L. Lawson, G. K. Soper, L. L. Wood, and J. B. Woodard, "Report of the commission to assess the threat to the United States from electromagnetic pulse (EMP) attack, critical national infrastructures," Apr. 2008.

[15] L. K. Gallos, R. Cohen, P. Argyrakis, A. Bunde, and S. Havlin, "Stability and topology of scale-free networks under attack and defense strategies," *Phys. Rev. Lett.*, vol. 94, no. 18, 2005.

[16] P. R. Goundan and A. S. Schulz, "Revisiting the greedy approach to submodular set function maximization," *Working paper*, 2008.

[17] M. M. Hayat, J. E. Pezoa, D. Dietz, and S. Dhakal, "Dynamic load balancing for robust distributed computing in the presence of topological impairments," *Wiley Handbook of Science and Technology for Homeland Security*, 2009.

[18] D. Hochbaum and A. Pathria, "Analysis of the greedy approach in problems of maximum k-coverage," *Naval Research Logistics (NRL)*, vol. 45, no. 6, pp. 615–627, 1998.

[19] IETF Internet Working Group, "Inference of Shared Risk Link Groups," November 2001, Internet Draft. [Online]. Available: http://tools.ietf.org/html/draft-many-inference-srlg-02

[20] Level 3 Communications, Network Map. [Online]. Available: http://www.level3.com/interacts/map.html

[21] G. Liu and C. Ji, "Scalability of network-failure resilience: Analysis using multi-layer probabilistic graphical models," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 319 –331, Feb. 2009.

[22] A. Narula-Tam, E. Modiano, and A. Brzezinski, "Physical topology design for survivable routing of logical rings in WDM-based networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 8, pp. 1525–1538, Oct. 2004.

[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - i," *Math. Prog.*, vol. 14, no. 1, pp. 265–294, December 1978.

[24] S. Neumayer and E. Modiano, "Network reliability with geographically correlated failures," in *Proc. IEEE INFOCOM*, Mar. 2010.

[25] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of the fiber infrastructure to disasters," in *Proc. IEEE INFOCOM*, Apr. 2009.

[26] ——, "Assessing the impact of geographically correlated network failures," in *Proc. IEEE MILCOM*, Nov. 2008.

[27] C. Ou and B. Mukherjee, *Survivable Optical WDM Networks*. Springer-Verlag, 2005.

[28] C. A. Phillips, "The network inhibition problem," in *Proc. ACM STOC*, 1993.

[29] A. Pinar, Y. Fogel, and B. Lesieutre, "The inhibiting bisection problem," in *Proc. ACM SPAA*, Jun. 2007.

[30] Qwest, Network Map. [Online]. Available: http://www.qwest.com/largebusiness/enterprisesolutions/networkMaps/

[31] A. Sen, S. Murthy, and S. Banerjee, "Region-based connectivity: a new paradigm for design of fault-tolerant networks," in *Proc. IEEE HPSR*, 2009.

[32] A. Sen, B. Shen, L. Zhou, and B. Hao, "Fault-tolerance in sensor networks: a new evaluation metric," in *Proc. IEEE INFOCOM*, 2006.

[33] M. Sharir and P. Agarwal, *Davenport-Schinzel Sequences and their Geometric Applications*. Cambridge University Press, 1995.

[34] J. Spragins, "Dependent failures in data communication systems," *IEEE Trans. Commun.*, vol. 25, no. 12, pp. 1494 – 1499, Dec. 1977.

[35] A. Vigneron, "Geometric optimzation and sums of algebraic functions," in *Proc. ACM-SIAM SODA*, 2010.

[36] C. Wilson, "High altitude electromagnetic pulse (HEMP) and high power microwave (HPM) devices: Threat assessments," CRS Report for Congress, July 2008. [Online]. Available: http://www.ntia.doc.gov/broadbandgrants/comments/7926.pdf

[37] W. Wu, B. Moran, J. Manton, and M. Zukerman, "Topology design of undersea cables considering survivability under major disasters," in *Proc. WAINA*, May 2009.

[38] XO Communications, Network Map. [Online]. Available: http://www.xo.com/about/network/Pages/maps.aspx

Authorized licensed use limited to: Hebrew University. Downloaded on June 17,2023 at 07:06:17 UTC from IEEE Xplore. Restrictions apply.