The Earth is nearly flat: Precise and approximate algorithms for detecting vulnerable regions of networks in the plane and on the sphere

Balázs Vass¹ | László Németh² | János Tapolcai¹

¹MTA-BME Future Internet Research Group and MTA-BME Information Systems Research Group, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, 1111, Hungary
²Department of Probability Theory and Statistics, Eötvös Loránd University, Budapest, 1111, Hungary

Correspondence
Balázs Vass, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest 1111, Hungary.
Email: vb@tmit.bme.hu

Funding information
This research was supported by the High Speed Networks Laboratory (HSNLab) hosted by the Budapest University of Technology and Economics. Emberi Erőforrások Minisztériuma, Grant/Award Number: BME FIKP-M/SC.
European Cooperation in Science and Technology, Grant/Award Number: CA15127. Országos Tudományos Kutatási Alapprogramok, Grant/Award Numbers: FK_17 123957, KH_18 129589, K_17 124171, and K_18 128062.

Abstract
Several recent works shed light on the vulnerability of networks against regional failures, which are failures of multiple pieces of equipment in a geographical region as a result of a natural disaster. To enhance the preparedness of a given network to natural disasters, regional failures and associated Shared Risk Link Groups (SRLGs) should be first identified. For simplicity, most of the previous works assume the network is embedded on a Euclidean plane. Nevertheless, they are on the Earth’s surface; this assumption causes distortion. In this work, we generalize some of the related results on the plane to the sphere. In particular, we focus on algorithms for listing SRLGs as a result of regional failures of circular or other fixed shape.

KEYWORDS
computational geometry, disaster protection of networks, failure modeling, shared risk link group, sphere, vulnerable region detection

1 | INTRODUCTION

Serious network outages are happening with increasing frequency due to disasters (such as earthquakes, hurricanes, tsunamis, tornadoes, etc.) that take down almost every equipment in a geographical area (see [9] for a recent survey conducted within COST Action RECODIS [21] on strategies to protect networks against large-scale natural disasters). Such failures are called regional failures and can have many locations, shapes, and sizes.

Due to the huge importance of telecommunication services, improving the preparedness of networks to regional failures is becoming a key issue [6, 8, 10–12, 16, 17, 19, 24]. Roughly speaking, protecting networks against regional failures is dealt with either by using geometric tools [1, 7, 19, 25–27, 29] or by reducing massively the problem space to a set candidate locations of failures [6, 10, 11, 16, 17, 24]. Nevertheless, both approaches require a detailed knowledge of the geometry of the network topology, such as the precise GPS coordinates of nodes and cable conduits’ routes, and the statistics of past disasters.

In many works, regional failures are computed by transforming the geographical coordinates of an existing network into a plane, which introduces distortion. Depending both on the geographical area of the network and on the transforming procedure,
this distortion can vary from negligible to significant. For example, the backbone network of a small-to-medium size country is not suffering a significant distortion when compared with the uncertainty of the available geographical data, but when turning to networks covering a large country, a continent, or even multiple continents, there is no projection which can hide the spherical-like geometry of the Earth’s surface (see Figure 1 taken from [23]). For example, while the territory of continental US can be mapped onto a plane with 4% distortion [23], if we want to investigate bigger networks, clearly there is no projection which can hide the spherical-like geometry of the Earth.

There are reasons why one should analyze the global communication network as a whole: Electromagnetic storms induced by the Sun’s coronal mass ejections could cause severe simultaneous failures of electric and communication networks all over the Earth1.

Backbone networks are designed to protect a given pre-defined list of failures, called Shared Risk Link Groups (SRLGs). Network recovery mechanisms are efficient if this SRLG list covers the most probable failure scenarios while having a manageable size.

An SRLG is called regional if it aims to characterize a failure damaging the network only in a bounded geographical area. It is still ongoing research on how to define and compute efficiently regional SRLG lists [25, 26, 29]. A common simplification of these works is that they compute the list of SRLGs on a planar representation of the networks; thus, our focus is to generalize these approaches to the sphere.

An SRLG consists of a set of network links, while node failures are implicitly defined (a node is considered to be failed with the SRLG if all its adjacent edges are part of the SRLG). If a failure \( f \) is listed as an SRLG, it is a common approach to skip listing any subset of \( f \) (if the network is protected to \( f \), it is also protected to any of its subsets) and, therefore, it is enough to list the maximal SRLGs caused by disasters.

Another issue is that the number of the listed SRLGs has to be kept low. With this aim, there is a common practice to fix the shape of the disasters [1, 27] with the assumption that every link intersecting the disaster area is destroyed, while the rest of the

1The magnetic storm of September 1 to 2, 1859 (a.k.a. the Carrington event) was the most intense in history. It was reported that during this electromagnetic storm, many fires were set by arcing from currents induced in telegraph wires (in both the United States and Europe). It is found that both the Carrington event solar flare energy and the associated coronal mass ejection speed were extremely high but not unique: some predict that the chance of a storm of this or even greater intensity in the next decade is 4 – 6% [15]. In a severe geomagnetic scenario like the Carrington event, an estimate of $1 trillion to $2 trillion during the first year alone was estimated as the societal and economic costs with recovery times of 4 to 10 years [3]. Electromagnetic storms even smaller than the Carrington event do affect today’s networks too: the outage in January 1994 of two Canadian telecommunication satellites during a period of enhanced energetic electron fluxes at geosynchronous orbit, disrupting telecommunication services nationwide. The first satellite recovered in a few hours; recovery of the second satellite took 6 months and cost $50 million to $70 million [3]
network is left intact. Among the possible geometric failure shapes, the most natural one is the circular disk, as it is compact, and is invariant to rotation. One possibility is to overestimate the possible failures with circular disks (or any other fixed geometric shape), which yields short SRLG lists. However, it is not clear what is the cost of this overestimation. In most of this work, we choose to overestimate the disasters by circular disks with a maximum size according to one among many possible measures, while we also briefly tackle overestimations with different geometric shapes.

When talking about (maximal regional) disk failures, the most natural measure is the disk radius, which represents the maximum geographical coverage of the natural disaster. Nevertheless, since the network density is usually not homogeneous (i.e., there are more nodes and links in crowded geographical areas than in non-crowded areas) the number of network elements (either nodes or links) contained by the disk are also two useful measures (it is natural that more SRLGs are needed in crowded areas and less crowded areas can be covered with fewer SRLGs). Therefore, in most of this work, we will concentrate on the following three types of SRLG lists:

- **maximal r-range SRLG list**: list of maximal link sets that can be hit by a disk with radius at most r.
- **maximal k-node SRLG list**: list of maximal link sets that can be hit by a disk hitting at most k nodes.
- **maximal k-link SRLG list**: list of maximal link sets that can be hit by a disk hitting at most k links.

To distinguish between lists obtained from planar and spherical representation, we will include attribute planar or spherical in the list names (e.g., maximal spherical r-range SRLG list) when needed for clarification.

It turns out that in all three mentioned cases, the size of the maximal SRLG list is linear in the network size in practice, and can be computed in low-polynomial time both in the planar case (maximal r-range list: [25], k-node: [29], k-link: [20]) and the spherical case.

This paper is an extension of [30], which is, to the best of our knowledge, the first study on enumerating regional SRLG lists in a spherical model. Also, in [30], the first comparison of the spherical and planar representations was provided. It is shown through examples that precise polynomial algorithms could be designed for spherical representation of the networks. In our experience, these algorithms are only 2 times slower than their planar pairs. We also believe that using our approach, resilient routing and network design results [14] can be further enhanced. Compared to the conference version of our paper [30], we have (1) included Section 4.2 presenting approximate algorithms on enumerating the maximal failures induced by arbitrary disaster shapes, (2) provided an enhanced mathematical analysis of algorithms presented Sections 4.1 and 4.2, and (3) in Section 5 added simulation results showing that the difference between the planar and spherical representations of the network can result in different SRLG lists even in case of networks having a geographic extension as small as 100 km.

As there are many mathematical derivations in the rest of the paper, we would like to summarize the concepts in plain text once again here for the sake of readability. As learned from previous studies, all of r-range, k-node, and l-link lists can be precisely calculated in low-polynomial time of the network size in case of planar representation of the network. Our first goal was to show that considering spherical embeddings of the networks the possibility of designing low-polynomial time algorithms for determining these SRLG lists remains. We demonstrated this phenomenon in Section 3 by designing an algorithm capable of determining the r-range SRLG list both in the planar and spherical case in low-polynomial time of the network elements. The existence of fast precise algorithms is good news, however, their drawback is that intuitively the faster they are, the harder they are to implement. This fact motivates our second goal, namely designing a framework of simple and fast algorithms capable of determining all the mentioned SRLG lists in both planar and spherical representation with enough precision, which are presented in Section 4.1. Our third and final goal is to show how simple approaches can be applied to more general models too: while in most of this study, we concentrated on disasters having a shape overestimated by a disk, Section 4.2 gives an outlook for designing easy-to-implement algorithms for essentially arbitrary disaster shapes.

The remaining of the paper is organized as follows. Section 2 describes the network representation model together with the assumptions made. In Section 3, we present an example of a polynomial algorithm for computing maximal SRLG lists, handling both the planar and spherical cases. While in Section 4.1, a faster and more flexible approximate approach is presented for solving the same problem, Section 4.2 gives an outlook for designing algorithms for arbitrary disaster shapes. Simulation results are presented in Section 5 and, finally, we draw the conclusions in Section 6.

## 2 MODEL AND ASSUMPTIONS

Throughout the paper, we will consider two types of embeddings of the network: embedding in Euclidean planar and spherical geometry.

The network is modeled as an undirected connected geometric graph $G = (V, E)$ with $n = |V| \geq 3$ nodes and $m = |E|$ edges stored in a lexicographically sorted list. The nodes of the graph are embedded as points in the Euclidean plane or sphere, and their precise coordinates are considered to be given in 2D and 3D Cartesian coordinate system in the planar and spherical
cases, respectively. Note that if coordinates are given in polar system (in case of spherical geometry), one can easily transform them to Cartesian at the very beginning.

When speaking of planar geometry, for each edge \( e \), there is a *polygonal chain* (or simply *polyline*) \( e' \) in the plane in which the edge lies (see Figure 2). Parameter \( \gamma \) will be used to indicate the maximum number of line segments a polyline \( e' \) can have. Naturally, in spherical case, the polyline of an edge refers to a series of geodesics. Note that this model covers special cases when edges are considered as line segments (geodesics).

It will be assumed that basic arithmetic functions \((+,\,-,\,\times,\,/,\,\sqrt{\cdot})\) have constant computational complexity. For simplicity, we assume that nodes of \( V \) and the corner points of the containing polygons defining the possible route of the edges are all situated in general positions of the plane, that is, there are no three such points on the same line and no four points on the same circle, and in the spherical case there are no antipodal nodes or breakpoints and no great circles of geodesics of polylines cross the North pole.

In this study, our goal is to generate a set of SRLGs, where each SRLG is a set of edges. Note that from the viewpoint of connectivity, listing failed nodes besides listing failed edges has no additional information. We consider SRLGs that represent worst-case scenarios the network must be prepared for and, thus, there is no SRLG which is a subset of another SRLG.

### 2.1 Model for circular disk-shaped disasters

In most of this study, it will be assumed that disasters are either having a shape of a circular disk or they are overestimated by a circular disk.

We will often refer to circular disks simply as disks. The disk failure model is adopted, which assumes that all network elements that intersect the interior of a circle \( c \) are failed, and all other network elements are untouched.

**Definition.** A circular disk failure \( c \) hits an edge \( e \) if the polyline of the edge \( e' \) intersects disk \( c \). Similarly node \( v \) is hit by failure \( c \) if it is in the interior of \( c \). Let \( E_c \) (and \( V_c \)) denote the set of edges (and nodes, respectively) hit by a disk \( c \).

We emphasize that in this model when we say \( e \) is hit by \( c \), it does not necessarily mean that \( e \) is destroyed indeed by \( c \); instead, it means that there is a positive chance for \( e \) being in the destroyed area. In other words, this modeling technique does not assume that the failed region has a shape of a disk, but overestimates the size of the failed region in order to have a tractable problem space.

**Definition.** Let \( C^p \) and \( C^s \) denote the set of all disks in the plane and the set of all disks on the sphere, respectively. For both geometry types \( g \in \{ p, s \} \), let \( C^r_g \), \( C^k_g \) and \( C^l_g \) denote the set of disks part of \( C^g \) having radius at most \( r \), hitting at most \( k \) nodes of \( V \) and hitting at most \( l \) links of \( E \), respectively.

Based on the above definition, we define the set of failure states that a network may face after a disk failure, with a maximal measure.

**Definition.** For all geometry types \( g \in \{ p, s \} \) and SRLG type \( t \in \{ r, k, l \} \), let set \( F(C^g_t) \) denote the set of edges which can be hit by a disk \( c \in C^k_t \), and let \( M_{t\text{max}}^g = M(C^k_t) \) denote the set of maximal edge sets in \( F(C^g_t) \).

Table 1 gives an overview on the corresponding notations and denominations of the SRLG list types we focus on this paper. Note that for every SRLG type \( t \in \{ r, k, l \} \) if \( f \in M_{t\text{max}}^g \) there is an \( f' \in M_{t\text{max}}^{g'} \) such that \( f \subseteq f' \) where \( s \leq s' \).

---

**FIGURE 2** Input graph \( G(V, E) \) with polylines, \( n = 17, \gamma = 4 \)
TABLE 1 Notations and denominations of the list types

| Notation | Denomination | Short name       |
|----------|--------------|------------------|
| \(M_p^r\) | maximal planar \(r\)-range SRLG list | planar \(r\)-range list |
| \(M_p^k\) | maximal planar \(k\)-node SRLG list | planar \(k\)-node list |
| \(M_p^l\) | maximal planar \(l\)-links SRLG list | planar \(l\)-link list |
| \(M_s^r\) | maximal spherical \(r\)-range SRLG list | spherical \(r\)-range list |
| \(M_s^k\) | maximal spherical \(k\)-node SRLG list | spherical \(k\)-node list |
| \(M_s^l\) | maximal spherical \(l\)-links SRLG list | spherical \(l\)-link list |

One aim of this study is to propose fast algorithms computing these lists for various sizes of \(m\).

2.2 Model for disasters with arbitrary shape

In Section 4.2, we give an overview of simple algorithms handling disasters which have a shape of a fixed non-self-intersecting closed polytope in the plane having boundaries consisting of line segments and arcs. The disaster can occur in any physical area and with any orientation. A link is hit if it has at least a common point with the disaster. The model is basically the same as the one described in the previous section, the only difference being the disaster shape.

3 PRECISE ALGORITHMIC APPROACHES FOR ENUMERATING MAXIMAL FAILURES

As mentioned before, determining the planar lists \(M_p^r\), \(M_p^k\), \(M_p^l\) is relatively well studied. It remains a question of how much the distortion of maps can affect the calculated SRLG lists. The answer is that it heavily depends on the projection used to make the map. For example, while the stereographic projection affects significantly the distances, but in contrast to many other projections it has the nice property of mapping spherical disks to planar disks [22] (fact also used in Appendix). One approach for calculating spherical lists would be to adapt existing algorithms to spherical geometry demonstrating the interoperability between these geometries. However, in this paper, we follow an approach simpler to present and avoiding trigonometric calculations via applying the projection in both directions for numerous times. In other words, some steps of the algorithm are performed on the plane, while others on the sphere.

In the following, we extend a precise algorithm for determining \(M_p^r\) (see [25]) to an algorithm computing \(M_p^r\) or \(M_s^r\) depending on the geometry of the input. In the rest of this section, we present this extended algorithm.

3.1 Smallest enclosing disks

Let us make the following definition for the sake of clarifying the intuition.

**Definition.** Let a disk \(c\) be smaller than disk \(c_0\), if \(c\) has a smaller radius than \(c_0\), or if they have an equal radius and the center point of \(c\) is lexicographically smaller than the center point of \(c_0\). Among a set of circles \(S_c\), let \(c\) be the smallest if it is smaller than any other circle in \(S_c\).

**Definition.** Let \(F \subseteq E\) be a finite nonempty set of edges (not necessarily a failure). We denote the smallest disk among the disks enclosing the polylines of \(F\) by \(c_F\) and we say \(c_F\) is the smallest enclosing disk of \(F\).

It is not difficult to see that \(c_F\) always exists for line segments or geodesics (depending on the geometry), and thus, by mapping the corresponding segments/geodesics together we can deduce that the definition is correct for polylines too. The key idea of our approach is that we can limit our focus only on the smallest enclosing disks \(c_F\). The consequence of the next proposition is that the number of smallest enclosing disks \(c_F\) is not too large.

**Proposition 1.** Let \(H\) be a nonempty set of polylines of edges with smallest enclosing disk \(c_H\). Then there exists a subset \(H_0 \subseteq H\) with \(|H_0| \leq 3\) such that \(c_H = c_{H_0}\).

**Definition.** Let \(S\) denote the set of maximal edge sets hit by a smallest enclosing disk.

By Proposition 1 we have:
Corollary 2. \(|S| \leq \binom{m}{3} + \binom{m}{2} + m = \frac{m^3}{6} + \frac{5m}{6}\).

Lemma 3. Let \(H\) be a set of line segments in the plane or geodesics on the sphere, \(|H| \leq 3\). Then \(c_H\) can be determined in \(O(1)\) time.

The proof of the Lemma is relegated to the Appendix.

Theorem 4. Let \(H\) be a set of polylines of edges, \(|H| \leq 3\). Then \(c_H\) can be determined in \(O(\gamma^3)\) time.

Proof. First, unpack each polyline into the \(\leq \gamma\) line segments/geodesics it is consisting of. Then, for each element \(h_i\) in \(H\), pick a segment \(s_i\). For each triplet (couple) of segments calculate the smallest enclosing disk (which by Lemma 3 can be done in \(O(1)\) time), and lastly choose the smallest from among the resulting disks.

3.2 Polynomial algorithm for determining maximal failures

In this section, we repeat an extension of the basic algorithm provided by [25] which handles both spherical and planar inputs.

There are two key facts inspiring this algorithm. Firstly, based on Proposition 1:

Corollary 5. (of Proposition 1). For both \(g \in \{p, s\}\) and every \(f \in M_g^r\) there exist \(\{e_1, e_2, e_3\} \in f\) such that \(c_{\{e_1, e_2, e_3\}} = c_f\).

Secondly, according to Theorem 4, smallest enclosing disks can be computed in \(O(\gamma^3)\) time both in the plane and on the sphere. Based on these, Algorithm 1 is presented, which is a straightforward basic polynomial algorithm. Here, the key idea is to maintain a list \(M'\) of maximal failures detected so far while scanning through the link sets \(f\) covered by the smallest enclosing disks of at most 3 edges. If there is no \(f \in M'\) containing \(f\), then \(f\) is appended to \(M'\) and all \(f \in M'\) which are part of \(f\) are removed as presented in Algorithm 2. This process is called refreshing.

The following theorem gives a very loose bound on the complexity of calculating \(M_g^r\).

Theorem 6. Algorithm 1 computes \(M_g^r\) in \(O(m^3(\gamma^3 + m^4))\) time. \(M_g^r\) has \(O(m^3)\) elements.

Proof. Based on Proposition 1, the algorithm is correct, it is computing \(M_g^r\). There are \(O(m^3)\) smallest enclosing disks to calculate, each in constant time. We claim that for each disk the calculation time of refreshing \(M_g^r\) with the resulting failure (according to Algorithm 2) is \(O(m^3 + \gamma^3)\) in case of each disk, because after the computation of the smallest enclosing disk in \(O(\gamma^3)\) time and determining \(f\) in \(O(m)\) time, we require to compare \(f\) with \(O(m^3)\) link sets, and each comparison can be done in \(O(m)\) time.

3.3 On improved complexity bounds

The results from the previous section can be easily improved using parametrization and some computational geometric tricks.

| Algorithm 1: Determining maximal \(r\)-range SRLG lists |
|-----------------|-----------------|
| **Input:** \(G(V, E), r, \text{geometry } g\), coordinates of nodes and edge polylines |
| **Output:** \(M_g^r\) |
| **begin** |
| \(M_g^r := \emptyset\) |
| Store \(E\) as a list, |
| for \(i_1 \in \{1, \ldots, m\}\) do |
| for \(i_2 \in \{1, \ldots, m\}\) do |
| for \(i_3 \in \{1, \ldots, m\}\) do |
| \(c_{i_1, i_2, i_3} := c(t_{i_1}, \ell(i_2), \ell(i_3))\) |
| if radius of \(c_{i_1, i_2, i_3}\) is \(\leq r\) then |
| \(f := F(c_{i_1, i_2, i_3})\) |
| refresh \(M_g^r\) with \(f\) // as in Algorithm 2 |
| **return** \(M_g^r\) |
The first observation is that for any meaningful radius of the disk failure most of the network will remain intact. However, failures with the same radius taking place in a crowded area tend to take down more equipment than the ones in sparsely inhabited areas. This motivates the introduction of a graph density parameter:

**Definition.** For every \( r \in \mathbb{R}^+_0 \), let \( \rho_r \) be the maximum number of edges which can be destroyed by a disk with radius at most \( r \).

This \( \rho_r \) is considered to be small in case of small \( r \) values. Another observation is that there are not many more network edges than nodes. This is formalized in the upcoming Claim 1.

Informally speaking, we denote the set of crossing points of the edges by \( X \). A more formal definition follows.

**Definition.** Let \( X \) be the set of points \( P \) in the plane on which no node element of \( V \) lies and there exist at least 2 edges which have polylines having a finite number of common points crossing each other in \( P \). Let \( x = |X| \).

Despite the fact that on arbitrary graphs \( x \) can be even \( O(n^4) \), in backbone network topologies typically \( x \ll n \) because a node is usually installed if two cables are crossing each other. This gives us the intuition that \( G \) is “almost” planar, and thus it has few edges.

**Claim 1.** The number of edges in \( G \) is \( O(n + x) \). More precisely for \( n \geq 3 \) we have \( m \leq 3n + x - 6 \).

**Proof.** If \( G \) is embedded in the plane, do the following. Let \( G_0(V \cup X, E_0) \) be the planar graph obtained from dividing the polylines of edges of \( G \) at the crossings. Since every crossing enlarges the number of edges at least by two, \( |E_0| \geq m + 2x \). On the other hand, \( |E_0| \leq 3(n + x) - 6 \) since \( G_0 \) is planar. Thus \( m \leq |E_0| - 2x \leq 3n + x - 6 \).

If \( G \) is embedded in the sphere, we can project it to the plane with stereographic projection, repeat the former arguments then apply an inverse projection to the sphere.

A third trick relies on the fact that in practice \( |M_r^g| \) is \( O(n) \) (as presented in planar case in [25]), thus in Algorithm 2 typically there have to be done only \( O(n) \) comparisons. Thus, we introduce a third parameter:

**Definition.** Let \( \lambda \) be the maximum cardinality of the list of maximal failures detected so far in Algorithms 1, 3, and 4, respectively.

Combining the former three observations, a lower parametrized complexity can be achieved:

**Theorem 7.** Algorithm 1 computes \( M_r^g \) in \( O((n + x)^3(n + x + \lambda \rho_r + \gamma^3)) \) time.

**Proof.** Based on Proposition 1, the algorithm is computing \( M_r^g \). There are \( O(n + x)^3 \) smallest enclosing disks to calculate, each in constant time. We claim that for each disk the calculation time of Algorithm 2 is \( O((n + x)^3 \rho_r + \gamma^3) \) in case of each disk, because after the computation of the smallest enclosing disk in \( O(\gamma^3) \) and determining \( f \) in \( O(n + x) \) we require \( O(\lambda) \) comparisons of link sets (i.e., pairs of possible maximal failures), and each comparison can be done in \( O(\rho_r) \) time.
Corollary 9 will give more intuitive bounds on the running time of Algorithm 1.

Definition. Let $diam$ be the geometric diameter of the network.

Proposition 8. Based on simulation results from Section VI of [25], in case of backbone networks, $\rho_r$ is proportional to $\frac{r}{diam}$ in the interval $(0, \frac{diam}{2}]$, where $diam$ is the geometric diameter of the network.

Corollary 9. (corollary of Theorem 7). If both $x$ and $\lambda$ are $O(n)$, Algorithm 1 computes $M^x_r$ in $O(n^3(n\rho_r + \gamma^3))$ time. If, in addition, $\gamma$ is $O(1)$ and $\rho_r$ is $O(r/diam)$, Algorithm 1 computes $M^x_r$ in $O\left(n^4 \frac{r}{diam}\right)$ time.

Corollary 9 proposes that $M^x_r$ can be determined in quartic time of $n$ in practice. On the other hand, Algorithm 1 has its limits of speed: because of the three nested for-loops, it runs in $\Omega(n^3)$ time. In order to achieve better results, the algorithm would have to be changed. For the planar case, Reference [25] gives an algorithm which runs in $O(n + x)^2 \rho_r^2)$ time for $\gamma = 1$ (i.e., the edges are considered as line segments there). Furthermore, we are convinced that an algorithm with parametrized running time near-linear in network size could be achieved for determining $M^x_r$ (and also for determining $M^x_k$ and $M^0_k$, despite the fact that they can be computed based on very different theories). However, presenting this kind of algorithm would exceed the limits of this paper.

4 | APPROXIMATE ALGORITHMS AND IMPLEMENTATION ISSUES

It is always good to have fast precise polynomial algorithms for solving a given problem. However, this approach also has some disadvantages: (1) the lower complexity a precise algorithm for determining a maximal circular SRLG list has, the harder to implement and prove its correctness and complexity; (2) designing algorithms for computing different types of maximal SRLG lists need totally different mathematics. Moreover, in most cases, the available geographical data of networks are inaccurate. Adding this fact to the inconveniences of the precise approach results into the idea of designing some approximate algorithms that are able to compute these lists with enough precision.

The main idea behind this class of algorithms that instead of keeping the original shape and size of the disaster area and taking in count all the infinitely many possible disaster centers, we slightly overestimate the disaster, letting us detect all the same (or occasionally a bit larger) hit link sets while going through only a finite number of centers.

4.1 | Approximate algorithms for circular disk failures

In this section, we present an approximate approach suitable for computing all types of maximal SRLG lists defined in Section 2.

Definition. For a point $P$ (in the plane or on the sphere) and node $v \in V$, let the node-distance couple be $[v, d(v, P)]$, where $d(v, P)$ is the distance between $v$ and $P$. Let $e(P)$ be the list consisting of the link-distance pairs of all links $e \in E$, sorted according to the lexicographical order of the links. Let $e(P)_\text{hit}$ be the sorted list of links not further from $P$ than $r$.

Proposition 10. For a given point $P$, both $e(P)$ and $e(P)_\text{hit}$ can be computed in $O((n + x)\gamma)$ time.

Clearly, both node-distance lists and edge-distance lists can be determined quickly. The plan is to determine these lists for enough points which are also placed well enough to be able to determine the maximal SRLG lists based on these node-distance and edge-distance lists.

Definition. Let $P$ denote the set of points $P$ for which we want to construct the link-distance lists $e(P)$.

Let us stick to planar geometry for a moment. Intuitively, we can calculate $M^x_r$ by including the grid points of a sufficiently fine grid (let us say containing 1 km x 1 km squares) in $P$. On the sphere, we should choose a similar nice covering. It is possible that we have some extra short links, thus for calculating the $k$-node and $k$-link list we should include some extra points in $P$. For example, by adding some random points of polylines of each edge and some point near every node, we can solve this issue.

Definition. Let $d_P$ be the maximal distance of any geometric location from the (closed) convex hull of the geometric embedding of graph $G$ to the closest point of set $P$, that is, $d_P := \max_{r \in \text{conv}(G)} \min_{p \in P} \text{dist}(p, r)$. 
Algorithm 3: Approximate algorithm for determining the maximal $r$-range SRLG lists

\begin{algorithm}
\begin{algorithmic}[1]
\Require $\mathcal{G}(V, E), r, P,$ geometry type $g,$ coordinates of nodes and polylines of edges
\Ensure $M_f^r$
\Begin
\For {$P \in P$}
\State determine $\epsilon(P)_{\text{hit}}$
\If {$\epsilon(P)_{\text{hit}} \neq \emptyset$}
\State refresh $M_f^r$ with $\epsilon(P)_{\text{hit}}$ \text{// according to Algorithm 2}
\EndIf
\EndFor
\State return $M_f^r$
\End
\end{algorithmic}
\end{algorithm}

**Definition.** Taking two sets $E_1$ and $E_2$, we denote the relationship of the sets with $E_1 \supseteq E_2$ if and only if for all $e_2 \in E_2$ there exists an $e_1 \in E_1$, such that $e_1 \supseteq e_2$.

Algorithm 3 is an example approximate algorithm for determining $M_f^r$.

**Theorem 11.** The resulting list $H_f^r$ of running Algorithm 3 is determined by the algorithm in $O(|P|((n + x)\gamma + \lambda \rho_f))$ time. Furthermore, $M_f^r \supseteq H_f^r \supseteq M_r^{-d_p}$.

**Proof.** Regarding the complexity, for an element $P$ of $P$ we have to construct $\epsilon(P)_{\text{hit}}$, which can be done in $O((n + x)\gamma)$ time, then refresh the list of suspected maximal failures with $\epsilon(P)_{\text{hit}}$ in $O(\lambda \rho_f)$, since the list contains at most $\lambda$ ordered lists consisting of at most $\rho_f$ edges.

On the other hand, $M_f^r \supseteq H_f^r$ is immediate, since the algorithm investigates only a subset of disks with radius $r$, while for every point $t$ in the $r$-neighborhood of $\text{conv}(G)$, there exists a $p \in P$ such that disk $c(t, r - d_p) \subseteq c(p, r)$, yielding $H_f^r \supseteq M_r^{-d_p}$, from which the proof follows.

Using the fact that the shape of the disasters is a closed disk, we get the following corollary:

**Corollary 12.** $\lim_{d_p \to 0} H_f^r = M_f^r$, for any fixed network.

**Corollary 13.** (of Theorem 11). $M_f^r \supseteq H_f^r \supseteq M_r^{-d_p}$. Furthermore, if both $x$ and $\lambda$ are $O(n)$, the resulting list $H_f^r$ of running Algorithm 3 is determined by the algorithm in $O(|P|n(\gamma + \rho_f))$ time. If in addition, $\gamma$ is $O(1)$ and $\rho_f$ is $O(r/\text{diam})$, $H_f^r$ is determined by Algorithm 3 in $O\left(|P|n \frac{1}{\text{diam}}\right)$ time.

Based on Theorem 11, if one wants to protect disasters caused by disks with radius $r$, it is only necessary to run Algorithm 3 initializing the radius as $r + d_p$.

Comparing Corollaries 13 and 9, we can see that despite the fact that the approximate Algorithm 3 is much simpler to implement, and taking into account that disasters are not precisely circular and the chosen radius is arbitrary, it clearly outperforms the precise Algorithm 1.

### 4.2 Approximate algorithms for disasters with arbitrary shape

Understanding how to deal with circular disk failures is a good start; however, one should consider other disaster shapes too. In [28], it is proven that, in a similar problem setting (the model of [28] considers that a link is hit by a disaster exactly if at least one of its endpoints is inside the disaster region. We believe our failure model is more realistic), there are a polynomial number of maximal failures caused by disasters having elliptic or polygonal (e.g., rectangular or equilateral triangular) shape. Again, as engineering fast precise algorithms for determining SRLG lists similar to $M_r, M_k$, or $M_f$ but for arbitrary disaster shapes instead of a circle is not trivial, we are going to discover the possibilities of approximate algorithms similar to the one described for determining $M_r$. In short, while the disk is invariant to rotation, now one should consider also the different orientations of the fixed shape. We shall discretize the continuous rotation of the shapes via taking only $a$ possible orientations of the shape:

**Definition.** Let $F_r$ be a set of non-self-intersecting closed polytopes embedded in the plane having boundaries consisting of line segments and arcs, for which (1) every $f_1, f_2 \in F_r, f_1$ can be moved into $f_2$ via a translation and a rotation, and (2) the smallest fitting disk of any $f \in F_r$ has a radius of $r$.

Let $N_{d_p}(F_r)$ consist of the elements $f$ of $F_r$, each $f$ extended (as $f' \coloneqq N_{d_p}(f)$) with the smallest $d_p$-neighborhood of $f$. 

Algorithm 4: Approximate algorithm for determining $M^F_p$ (the maximal failures the disaster shape $f$ can cause)

**Input:** $G(V,E), r, F_r, a, P$, geometry type $g$, coordinates of nodes and polylines of edges  

**Output:** $M^F_p$耐

\begin{algorithmic}
\STATE \textbf{begin}
\STATE Calculate $d_P$
\STATE \textbf{Let} $f \in F_r$, \textbf{calculate} $f''$ \textbf{according to the definition of} $N(F_r)$
\FOR{$P \in P$} do
\STATE translate $f''$ \textbf{such that} $T = P // T$ \textbf{is a fixed interior point of} $f''$
\FOR{$z \in \{0, \frac{2\pi}{\gamma_2}, \ldots, \frac{2\pi(z-1)}{\gamma_2}\}$ do
\STATE $f_z := \text{rotate}(f'', T, z)$
\STATE refresh $M^F_{P,F_r}$ with $\{t \in E | f_z \cap \text{polyline of } e \neq \emptyset\} // \text{according to} \text{Algorithm 2}$
\RETURN $M^F_{P,F_r}$ \textbf{as} $M^F_p$, \textbf{i.e.}, the maximal failures the disaster shape $f$ can cause
\STATE \textbf{end}
\STATE \textbf{end}
\end{algorithmic}

Let $T$ be an arbitrary point of a polytope $f \in F_r$. Let $N(F_r) = N_{d_p+d_{a,T}}(F_r)$ be the consisting of the elements $f'$ of $N_{d_p}(F_r)$, each $f'$ extended as $f''$ with the smallest $d_{a,T}$-neighborhood of $f'$ (where distance $d_{a,T}$ is a function of the extended shape $f'$, the number of orientations of the shape considered $a$, and the center of rotation $T \in \text{int}(f)$) in such a way that:

$$\bigcup_{z \in [0,2\pi]} \text{rotate}(f', T, z) \subseteq \bigcup_{z \in [0,2\pi]} \text{rotate}(f'', T, z). \tag{1}$$

By the former definition, we have the following. For any possible place and orientation of a disaster with shape $f$, in which $f$ intersects at least a network element, there exists a $p \in P$ and an orientation of the extended disaster shape $f''$ for which the area destroyed by $f''$ contains the area destroyed by $f$. This also holds if the orientation is taken from among the $a$ investigated orientations. Thus the edge set hit by $f''$ is a superset of the edge set hit by $f$.

**Proposition 14.** $\lim_{d_p \to 0} (d_P + d_{a,T}) = 0$.

**Definition.** Let $\phi$ denote the number of line segments and arcs needed to describe a shape from $F_r$. Let $\phi_r$ be the maximal number of edges an $f \in F_r$ can hit.

**Definition.** For geometry type $g \in \{p, s\}$, let denote the set of maximal failures caused by disasters from $F_r$ and $N(F_r)$ by $M^F_{F_r}$ and $M^{g}_{N(F_r)}$, respectively.

Algorithm 4 is an example approximate algorithm for determining $M^F_p$.

**Theorem 15.** The resulting list $M^F_{N(F_r)}$ of running Algorithm 4 is determined by the algorithm in $O\left( |P| a \left( \phi + (n + x) \left( \phi + \frac{\log |P|}{a} \right) \gamma + \lambda \phi_r \right) \right)$ time if the original failure shape $f$ is convex. Furthermore, for any given network, $\lim_{d_p \to 0} M^F_{N(F_r)} = M^F_p$.

**Proof.** Clearly, the algorithm computes $M^F_{N(F_r)}$. Based on Proposition 14 proposing that as $d_P \to 0$ and $a \to \infty$, the necessary inflation of the original shape $f$ tends to 0, we also can see that for any given network, $\lim_{d_p \to 0} M^{g}_{N(F_r)} = M^F_p$.

Regarding its complexity, we can argue as follows.

To calculate $d_P$, we observe that all the points in $N_r(G)$ (the smallest $r$-neighborhood of the convex hull of $G$) maximizing the distance from $P$ lie on the intersection of at least 3 Voronoi cells computed on point set $P$, restricted to $N_r(G)$, and the outer region of $N_r(G)$ taken as an extra cell. The Voronoi regions of $P$ can be computed in $O(|P| \log |P|)$ time both in the plane and on the sphere (plane: Section 7.2 in [5], sphere: special case of Corollary 2 in [18]). Using Claim 1, one can see that the number of line segments and arcs (or geodesics on the sphere) needed to describe $N_r(G)$ is $O(n x \gamma)$, thus we claim the total complexity of computation and description the modified Voronoi diagram described before is $O(n x \gamma |P| \log |P|)$. Based on the graph representation of the diagram one can determine $d_P$ in the proposed complexity.

We claim that Line 2 can be done in $O(\phi)$ time if $f$ is convex. In case of non-convex failure shapes, $f$ calculations can become more complicated (e.g., holes in $f$ can disappear in $f''$), but clearly, $f''$ can be determined in time polynomial in $\phi$ in this case too.

For a given $P \in P$, translation of $f$ (Line 4) can be done in $O(\phi)$ time.
For a given \( P \in \mathcal{P} \) and \( z \in \left\{ 0, \frac{\pi}{a}, \ldots, \frac{2(\lambda-1)\pi}{a} \right\} \), rotation (Line 6) can be done in \( O(\phi) \) time.

For a given point and direction, Line 7 needs \( O((n+x)\phi \gamma + \lambda \phi \gamma) \) operations, as follows. For every edge \( e \in E \) one has to check whether its polyline \( e' \) intersects the boundary of \( f \); or if not, is \( e' \) situated entirely in \( f \), both can be checked in \( O(\phi \gamma) \). Then, refreshing \( M^r_{N(F)} \) with the edge set hit by \( f \), can be done in \( O(\lambda \phi \gamma) \) time, since edges are stored in ordered lists.

We can conclude that Algorithm 4 is correct and runs in the proposed complexity.

Algorithm 4 clearly runs in time polynomial in the input size (even for non-convex disaster shapes \( f \)), and can be applied for a wide variety of disaster shapes.

Corollary 17 will offer a more intuitive complexity result on Algorithm 4:

**Proposition 16.** Since any \( f \in F \) can be covered with a disk having a radius \( r \), \( \varphi \leq \rho \). Based on Proposition 8, this also means that \( \varphi \) is \( O \left( \frac{r}{\text{diam}} \right) \) in case of backbone networks.

**Corollary 17.** If the number of edge crossings \( x \) is \( O(n) \), parameters \( \gamma \) and \( \phi \) are \( O(1) \), and \( \log |P| \) is chosen to be \( O(a) \), the resulting list \( M^r_{N(F)} \) of running Algorithm 4 is determined in \( O(|P| a(n + \lambda \phi \gamma)) \) time. If, in addition, \( \lambda \) is \( O(n) \), and \( \varphi \) is \( O \left( \frac{r}{\text{diam}} \right) \), the runtime is \( O \left( a|P| n \frac{r}{\text{diam}} \right) \).

If we compare Corollaries 17 and 13, we can see that unless the shape \( f \) of disasters is very complicated, the only real additional complexity compared to the circular disk failure case arises from the fact that that usually \( f \) is not invariant to rotation.

Comparing Corollaries 17 and 9, we can see that although the approximate Algorithm 3 handles the much more complex problem induced by disasters having an arbitrary given shape \( f \), it has a lower complexity compared to the precise Algorithm 1 dealing with only circular disasters, showing the strength of the proposed framework of approximate algorithms.

## 5 | SIMULATION RESULTS

In this section, we present numerical results that validate our approximate approach for circular disk failures presented in Section 4.1, and demonstrate the use of the proposed algorithms on some realistic physical networks. The algorithm was implemented in Python 3.5 using various libraries. Distance functions were implemented from scratch. No special efforts were made to make the algorithm space or time optimal. Run-times were measured on a commodity laptop with Core i5 CPU at 2.3 GHz with 8 GB of RAM. The output of the algorithm is a list of SRLGs such that no SRLG contains the other.

We interpret the input topologies in two ways: polygon, where links are polygonal chains, and line, where the corner points of the polygonal links are substituted with nodes (of degree 2). Here links are line segments.

### 5.1 | Extreme geographical extension makes difference

We found that running times for spherical representations were \( \sim 2 \) times slower than the planar ones in case of most networks (see Table 2). The only exception is when the network has an extreme geographic extension (e.g., AboveNet), in this case, the obtained SRLG lists tend to be longer (Figure 3 demonstrates this in case of \( k \)-link lists) causing a slight increase both in the parameter \( \lambda \) and in the running time.

Another issue which can be noticed is related to the achievable preciseness using the approximate approach. Based on Theorem 11, running time is proportional to \( |P| \); given this and the running times collected in Table 2, we can deduce that if the drop of price of computation power remains for an additional short time period, one will be able to run these simulations.

| Name          | \( |V| \) | \( |E| \) | Planar runtime | Spherical runtime |
|---------------|--------|--------|----------------|------------------|
|               | Polygon | Line   | Polygon | Line | Polygon | Line | Polygon | Line |
| AboveNet      | 9       | 22     | 15      | 28   | 232     | 156  | 410     | 757  |
| LambdaNet     | 10      | 33     | 10      | 33   | 282     | 225  | 444     | 410  |
| GARR (Italy)  | 16      | 16     | 18      | 18   | 107     | 92   | 204     | 187  |
| GTS (Hungary) | 14      | 15     | 39      | 26   | 175     | 146  | 311     | 291  |
When can we neglect the curvature of the Earth?

An important question is that, in practice, under which geographic extension of the network can one say that, in the viewpoint of SRLG enumeration, it is practically indifferent whether we consider a spherical or a planar representation of the network.

5.3 When can we neglect the curvature of the Earth?

An important question is that, in practice, under which geographic extension of the network can one say that, in the viewpoint of SRLG enumeration, it is practically indifferent whether we consider a spherical or a planar representation of the network.
FIGURE 4 Example of list behaviors: 28_optic_EU (|V| is 19 and 28, |E| is 32 and 41, in polygon and line case, respectively). A, |M^r_1| compared to |M^r_1| for all possible l/k/r values. B, |M^r_1| compared to |M^r_1| and |M^r_1| for small l/k/r values. C, A planarization of 28_optic_eu.

FIGURE 5 The ratio of those SRLGs which are different in M^r_1 and M^r_1, that is, |M^r_1 \triangle M^r_1|/|M^r_1| \cup |M^r_1|

In other words, focusing now only on lists M^r_1, the question is that under which size of the physical network will M^r_1 and M^r_1 (maximal link sets which can be hit by a single circular disk with radius r, in the plane and on the sphere, respectively) be the precisely the same. The answer depends not only on the physical size, but also on the characteristics of the network itself: it can represent a dense metropolitan backbone network with multiple nodes close to each other, but it can also be geographically very sparse. Keeping in mind that in metropolitan areas there can be many more differences, we took network AboveNet (Figure 3C), and its shrunk instances, where AboveNet/c means that we rotated AboveNet such that the average lat and lon coordinates are both 0, then we divided each coordinate by c. We were interested in the smallest possible shrinking ratio, for which M^r_1 = M^r_1.

We used M(r) := |M^r_1 \triangle M^r_1|/(|M^r_1| + |M^r_1|) \in [0, 1] (the ratio of SRLGs, which are present in only one of M^r_1 and M^r_1) as a similarity measure: if M(r) is close to 1, it means the two lists are very different, while if it is close to 0, it means there are few differences. We set r = 8 to be a bit larger than half of the diameter of the current network, r = 0 to be a small radius, the rest of the r values were linearly interpolated.

Figure 5 shows that, while in case of AboveNet, M^r_1 and M^r_1 are almost entirely different for many values of r, the tendency is that M(r) decreases as the physical size of the network decreases, which nicely fits the intuition. Surprisingly, M(r) is not 0 for every range r even for AboveNet/300, which corresponds to the case when the approximate network diameter is 104 km, AboveNet/400 (having a diameter of approx. 74 km) being the most spread out instance where M^r_1 and M^r_1 are the same for all investigated r ranges.
As a rule of thumb, we can deduce that the difference between the planar and spherical representations of the network can result in different SRLG lists even in the case of networks having a geographic extension as small as 100 km.

6 | CONCLUSION

We investigated the problem of generating SRLG lists of networks. We found that the known precise low-polynomial SRLG generating techniques can be modified in order to fit the spherical geometry, allowing us to generate SRLG lists with more precision. A framework of easy-to-implement approximating algorithms for determining the SRLG lists in both planar and spherical representation was also presented.

In our experience, SRLG lists generated using the spherical representation of the networks are different from the planar ones, and also they tend to be longer, especially in case of extreme geographical extension. In the case of our implementation, enumerating SRLG lists in case of spherical representations was typically 2 times slower than in the planar case. The difference between the planar and spherical representations of the network can result in different SRLG lists even in the case of networks having a geographic extension as small as 100 km.

While in most of our study, we supposed the disasters destroy the network in an area of a circular disk, some results were generalized to essentially arbitrary disaster shapes.

ORCID

Balázs Vass https://orcid.org/0000-0002-8589-7165
János Tapolcai https://orcid.org/0000-0002-3512-9504

REFERENCES

[1] 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 21 (2013), 1525–1538.
[2] G. Coaklay, Analytical solutions of the ten problems in the tangencies of circles; and also of the fifteen problems in the tangencies of spheres, Math. Monthly 2 (1860), 116–126.
[3] N.R. Council, Severe Space Weather Events: Understanding Societal and Economic Impacts: A Workshop Report, The National Academies Press, Washington, DC, 2008.
[4] H.S.M. Coxeter, The problem of Apollonius, Am. Math. Monthly 75 (1968), 5–15.
[5] M. De Berg, M. Van Kreveld, M. Overmars, and O.C. Schwarzkopf, Computational Geometry, Springer, 2000.
[6] F. Dikbiyik, M. Tornatore, and B. Mukherjee, Minimizing the risk from disaster failures in optical backbone networks, J. Lightwave Technol 32 (2014), 3175–3183.
[7] M.T. Gardner and C. Beard, Evaluating geographic vulnerabilities in networks, 2011 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), May 2011, pp. 1–6.
[8] O. Gerstel, M. Jinno, A. Lord, and S.J.B. Yoo, Elastic optical networking: A new dawn for the optical layer? Commun. Magazine IEEE 50 (2012), s12–20.
[9] T. Gomes, J. Tapolcai, C. Esposito, D. Hutchison, F. Kuipers, J. Rak, A. de Sousa, A. Iossifides, R. Travancas, J. André, L. Jorge, L. Martins, P.O. Ugalde, A. Pašić, D. Pezaros, S. Jouet, S. Secci, and M. Tornatore, A survey of strategies for communication networks to protect against large-scale natural disasters, 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM), Sept 2016, pp. 11–22.
[10] M.F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee, Design of disaster-resilient optical datacenter networks, J. Lightwave Technol 30 (2012), 2563–2573.
[11] I.B.B. Harter, D. Schupke, M. Hoffmann, G. Carle, et al., Network virtualization for disaster resilience of cloud services, Commun. Magazine IEEE 52 (2014), 88–95.
[12] J. Heidemann, L. Quan, and Y. Pradkin, A Preliminary Analysis of Network Outages During Hurricane Sandy, Information Sciences Institute, University of Southern California, Los Angeles, CA, 2012.
[13] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, The internet topology zoo, IEEE J. Sel. Areas Commun. 29 (2011), 1765–1775.
[14] A. Kushwaha, D. Kapadia, A. Gumaste, and A. Somani, Designing multi-layer provider networks for circular disc failures, International Conference on Optical Network Design and Modeling (ONDM), Dublin, Ireland, May 2018.
[15] G.S. Lakhina and B.T. Tsurutani, Geomagnetic storms: Historical perspective to modern view, Geosci. Lett 3 (2016), 5.
[16] X. Long, D. Tipper, and T. Gomes, Measuring the survivability of networks to geographic correlated failures, Opt. Switching Netw 14 (2014), 117–133.
[17] B. Mukherjee, M. Habib, and F. Dikbiyik, Network adaptability from disaster disruptions and cascading failures, Commun. Magazine IEEE 52 (2014), 230–238.
[18] H.S. Na, C.N. Lee, and O. Cheong, Voronoi diagrams on the sphere, Comput. Geom 23 (2002), 183–194.
[19] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, Assessing the vulnerability of the fiber infrastructure to disasters, IEEE/ACM Trans. Netw. 19 (2011), 1610–1623.
[20] E. Papadopoulou and M. Zavershynskyi, The higher-order Voronoi diagram of line segments, Algorithmica 74 (2016), 415–439.
[21] J. Rak, D. Hutchison, E. Calle, T. Gomes, M. Gunkel, P. Smith, J. Tapolcai, S. Verbrugge, and L. Wosinska, RECODIS: Resilient communication services protecting end-user applications from disaster-based failures, 18th international conference on transparent optical networks (ICTON), July 10-14 2016, Invited Paper.
[22] B. Rozenfeld and N.D. Sergeeva, Stereographic Projection, Mir Publisher, Moscow, 1977.
APPENDIX: DETERMINING SMALLEST ENCLOSED DISK OF LINE SEGMENTS OR GEODESICS IN $O(1)$ TIME

Proof of Lemma 3. For planar geometry, this problem is already solved, see Theorem 3 of paper [25]. It remains to prove it in case of spherical embedding.

Let $e_1$, $e_2$, $e_3$ be three geodesics on the sphere. The endpoints $(p_{i1}, p_{i2})$ are given by Cartesian coordinates $(x_{i1}, y_{i1}, z_{i1})$, $(x_{i2}, y_{i2}, z_{i2})$. Let $p_{i3}$ be an arbitrary point inside $e_i$. Points $p_{i1}, p_{i2}$ and $p_{i3}$ determine the great circle on the sphere containing geodesic $e_i$.

We will project geometric objects on the sphere to the plane using the stereographic projection from the north pole, which has the property that the image of a spherical circle will be a circle on the plane, or in special case, if it contains the north pole, its image is a line [22]. Note that for the sake of simplicity it is assumed that no great circle investigated crosses the north pole.

Projecting the 9 spherical points onto the plane, we receive $q_{ij}$ points given by Cartesian coordinates $(x_{ij}, y_{ij}, z_{ij}) \mapsto q_{ij} = \left(\frac{y_{ij}}{1-z_{ij}}, \frac{x_{ij}}{1-z_{ij}}\right)$. We denote the images of $e_1$, $e_2$, $e_3$ by arcs $f_1, f_2, f_3$. Calculating the radius and center point of the containing circle $c_i$ for arc $f_i$ requires constant number of coordinate geometric steps. Let $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and $r_1, r_2, r_3$ be the Cartesian coordinates of the center of containing circles, and radii, respectively.

The smallest enclosing disk $c_H$ on the sphere has an image $c'_H$ on the plane. However, the parameters of $c'_H$ are different from the parameters of $c_H$, and the images of the fitting points of $c_H$ and $c'_H$, are the fitting points of $c'_H$ and $f_i$. That inspires the plan to find all the fitting circles of $f_i$ (i.e., those which have exactly one common point with each $f_i$ or which have one common point with two of them and containing the third) on the plane, project them back onto sphere and select the smallest among them, as that is the smallest enclosing disk of $e_1, e_2, e_3$. Thus, we need to find the potential best fitting circles in the plane.

It is possible that the disk fits for two arcs and includes some points of the third. We can choose two arbitrary arcs in 3 ways. Choosing $f_1, f_2$ we must calculate the distance of the two arcs and use it as the diameter of the potential disk. On each arc, the distance is determined by an inside point or one of the boundary points. Calculating the distance of two points, a point and a circle or two circles have both constant complexity. So, in this case, 3 · 32 · $O(1)$ calculations are required.

If the smallest disk touches all of the arcs there are also more different cases. Each arc can be touched on a boundary point or on an inside point (33 cases). Fortunately, fitting a circle is already solved in all of the cases and these are known as the problems of Apollonius [4].

If the smallest enclosing disk touches all three arcs $f_1, f_2$, and $f_3$, we have three cases for each arc $f_i$: the disk either touches the arc in an interior point or at one of its endpoints. In the former case let $(x_1, y_1)$ and $r_1$ be the Cartesian coordinates of the center point and radius of the containing circle of arc $f_1$, respectively. In the latter case, let $(x_1, y_1)$ be...
coordinates of the endpoint itself, while let \( r_i \) be 0. Numbers \( s_1, s_2, \) and \( s_3 \) are \( \pm 1 \) representing that the fitting circle touches on the outside or on the inside of the containing circles of \( c_1, c_2, \) and \( c_3 \) (2³ different possibilities to be checked in each case). Parameters \( x_s, y_s, \) and \( r_s \) of the fitting \( C \) circle can be calculated by solving the following system of equations [2]:

\[
\begin{align*}
(x_s - x_1)^2 + (y_s - y_1)^2 &= (r_s - s_1 \cdot r_1)^2 \\
(x_s - x_2)^2 + (y_s - y_2)^2 &= (r_s - s_2 \cdot r_2)^2 \\
(x_s - x_3)^2 + (y_s - y_3)^2 &= (r_s - s_3 \cdot r_3)^2.
\end{align*}
\]

The system is quadratic, thus it can be solved in a constant number of arithmetic calculations. The complexity of these calculations all together are \( 3^3 \cdot 2^3 \cdot O(1) \).

After finding the \( 3^3 \cdot 2^3 + 3 \cdot 3^2 \) possible smallest disks, we must project them back to the surface of the sphere. We are allowed to use only the two endpoints of an arbitrary diameter from each possible circle. This requires \( 2 \cdot 3^5 \) coordinate transformations \((x, y) \rightarrow \left( \frac{2x^2}{1+x^2+y^2}, \frac{2y^2}{1+x^2+y^2}, -\frac{1+2x^2+y^2}{1+x^2+y^2} \right)\).

Finding the minimal radius of potential disks requires \( 3^5 - 1 \) comparisons between diameters. Using this method the minimal disk for \( e_1, e_2, e_3 \) can be determined in \( O(1) \) time. However, the algorithm could be improved by using pre-computations for the edges, excluding some possible disks already on the plane instead of transforming back or fixing \( s_i \) in case of boundary points. Note that only a constant number of basic arithmetic functions (+, −, ×, ÷, √) were used during the computation.