Enumerating Maximal Shared Risk Link Groups of Circular Disk Failures Hitting *k* Nodes

Balázs Vass*, Erika Bérczi-Kovács†, János Tapolcai*
*MTA-BME Future Internet Research Group, Budapest University of Technology and Economics, {vb,tapolcai}@tmit.bme.hu
†MTA-ELTE Egerváry Research Group on Combinatorial Optimization, and
Department of Operations Research, Eötvös Loránd University, Budapest, Hungary, koverika@cs.elte.hu

Abstract—**Many recent studies shed light on the vulnerability of networks against large scale natural disasters. The corresponding network failures, called regional failures, are manifested at failing multiple network elements that are physically close to each other. The recovery mechanisms of current backbone networks protect failures listed as Shared Risk Link Groups (SRLGs). The root of the problem is that the existing routing protocols were not designed to maintain any geographic information of the physical topology, such as the coordinates of the nodes and the routes of the cable conduits. We aim to design an algorithm for the routing engines which can generate a reasonable list of SRLGs based on the limited geometric information available. As a first step towards this direction, in this paper, we propose a limited geographic information failure model for the network topology, that enables efficient algorithms to compute the set of links which are expected to be close to each other. More precisely, we work with (1) relative node positions without knowing the real distances, (2) an area in the map defines the route of each physical cable, and (3) a regional failure is a circular disk with $k = 0, 1, \dots$ nodes in its interior. We describe an efficient algorithm for listing SRLGs based on our limited geographic information failure model, and show that under realistic assumptions the obtained list of SRLGs is short, approximately $1.2n$ and $2.2n$ for $k = 0$ and $k = 1$, respectively, where n is the number of nodes of the network.**

I. Introduction

Networks can efficiently survive the failure of a single network element(we refer the interested readers to [1], [2]); however, we are still witnessing severe network outages [1]–[11] because of a failure event that takes down almost every equipment in a physical region as a result of a disaster, such as weapons of mass destruction attacks, earthquakes, hurricanes, tsunamis, tornadoes, etc. These type of failures are called *regional failures*, which are simultaneous failures of nodes/links located in at most a few hundred kilometer wide geographic area [1]–[11]. The distance between the endpoints of most network connections are often thousands of kilometers; thus, the users are usually physically far from the natural disaster and may not be tolerant with a network outage. Therefore, increasing the service availability by allocating longer backup paths with more significant physical distance from the primary path is desired.

The current best practice is to ensure that the primary and backup paths assigned to a connection are node disjoint. Compared to edge-disjointness, in this way operators ensure that the distance between the nodes of the primary and backup paths (except at the terminal nodes) are at least 1-hop-distance from each other. The intuitive reasoning is that a link in a backbone network is typically a few hundred kilometers long, while natural disasters are never larger than a few hundred kilometers. The root of the outages is usually because: I) *close nodes* when two nodes are placed close to each other; for example, in highly populated areas. II) *parallel links* when two links are placed close to each other because of some geographic reasons; for example, they traverse the same bridge over a large river or cross a mountain range through the same valley.

A straightforward solution would be that at network planning, operators collect historical data about the vulnerabilities (e.g., past natural disasters) and combine it with the geometry of the network topology (e.g., the exact GPS coordinates of the nodes and the routes of the cable conduits). Several former studies investigated this problem, by either utilizing geometric approaches [3], [12]–[16] or aggressively reducing the problem space by identifying a few candidate locations of failures [5], [7]–[11]. Unfortunately, handling the geometric information with the network topology is not part of the current best practice, and the repositories about past natural disasters are far from complete. Furthermore, the Internet Service Provides usually hire the links as a service from an independent company, called the Physical Infrastructure Provider, and thus, operators have no information about the route of the links, or the physical coordinates of the intermediate routing nodes. These all translate to the best practice to prepare networks for single element failures only.

This paper steps back, and instead of focusing on the properties of the disaster and physical environment itself, we focus on adapting the ideas in computational geometry. We borrow the idea of Gabriel graphs to classify "close nodes": a node-pair i, j is close if the circular disk with diameter nodes i and j has no nodes interior. For "parallel links", inspired by the Voronoi diagrams [17], a set of links S is classified as close if there is a point in the plane from which the links of S are the nearest network elements, and thus there could be a disaster that will destroy all the links of S while the rest of the network remains operating. More precisely, our definition is based on the dual of the Voronoi diagram, called the Delaunay graph, which has the Gabriel graphs as sub-graph.

Inspired by the above, next we define a *limited geometric* *information failure model* to estimate the links that are potentially close to each other based on the logical topology and a little information a router may know about the network geometry. The model represents the outcome of a possible topology discovery software module running in the routing engines, which embedded the nodes and links into a schematic map according to the measured propagation delays. This study aims to design an algorithm for the routing engines which can generate a reasonable list of SRLGs based on the limited geometric information available.

A. The Limited Geometric Information Failure Model of the Network

We have the following design goals in defining the limited geometric information failure model.

(i) Do *not underestimate* the set of links involved in a possible regional failure. We believe the operator's damage in case of an unprotected regional failure is much higher than the extra cost of protecting networks against larger SRLGs.
(ii) *Relative link distances* are given, the exact route of the cables are unknown, and the nodes are embedded in a schematic map.
(iii) Provide a *fast and space efficient* way of calculating the set of SRLGs.

According to our first design goal, we deal with circular disk failures and define the size of the regional failure through the number of nodes it covers. Although the regional failures can have any location, size, and shape, without any background information on the regional failures, it is a common practice to overestimate the size of the regional failure by ignoring its shape and rather focus on its radius only [15], [18][1]. According to our second design goal, the scaling of the topology map is not known, thus we cannot define a fix maximum radius for the regional failures, but instead, we define a limit on the number of nodes interior to the circular disk.

Now we can define the limited geometric information failure model which is based on the following assumptions:

(1) The network is a geometric graph $\mathscr{G}(V,\mathscr{E})$ embedded in a 2D plane.
(2) The exact route of the conduits of the network links are not known, but contained by a polygonal region.
(3) The shape of the regional failure is assumed to be a circular disk with arbitrary radius and center position.
(4) We focus on *regional link k-node failures*, failures that hit k nodes for $k \in \{0, |V|-2\}$.

The detailed model description can be found in Sec. II. We argue this failure model can reasonably represent the possible regional failures, without actually requiring to know the scaling of the topology map.

Based on our output, operators can generate SRLG-disjoint primary and backup paths to protect the connection against natural disasters[2]. The distance between the primary and backup paths is a straightforward metric to compare the failure models. Based on the logical topology, the conventional approach to defining the distance is the hop-distance between the nodes traversed by the primary path and the nodes traversed by the backup path, except the terminal nodes. Based on this definition, we can list the failure models in increasing order of their strength (see Appendix A for the proof).

- Single link failures (≥ 0-hop-distance).
- Single node failures (≥ 1-hop-distance).
- Single regional link 0-node failures.
- Single regional link 1-node failures.
- Single regional link 2-node failures, etc.

Note that, in our experiments with practical network topologies protecting against single regional link 0-node failures resulted in at least 2-hop-distance between the nodes of the primary and backup paths, except the terminal nodes. We believe the proposed approach well captures the possible regional network failures based on the little geographic information available at network devices.

B. Survivable Network Design

Survivability against failures is a mandatory requirement in the design and operation of backbone networks. *Shared Risk Link Groups (SRLG)* (also referred to as *failure states*) are the abstraction model for defining sets of failures for which the network should be prepared. Note that SRLG is defined as a set of links, we say a node is part of an SRLG if every adjacent edge to the node is part of the SRLG. The backup resources are allocated in the network so that it can automatically recover every connection after the failure of a single SRLG. In practice, it is done by reconfiguring a bypass tunnel/lightpath around the failed set of nodes and links. In other words the network can survive a failure of an SRLG or a subset of links in an SRLG; however, there is no performance guarantee when a network is hit by a failure that involves links not a subset of an SRLG. Nevertheless, the list of SRLGs must be defined very carefully, because not getting prepared for one likely simultaneous failure event means a significant degradation in the observed reliability of the network.

While many recovery mechanisms can, in principle, load arbitrary lists of SRLGs, in practice, these lists currently only contain a single link or node failure. Here the concept is that the failure first hits a single network element for the protection of which the network is already pre-configured. In practice there are two criteria on the list of SRLGs:

(i) the size of an SRLG should be small, and
(ii) the number of SRLGs should be small.

Intuitively, the first is important because larger SRLGs require to reserve more protection bandwidth, the second is because network operators manually configure the list of SRLGs[3].

[1] Of course, extreme overestimation of the failed link set should be avoided.

[2] The routing algorithms modify the SRLGs, whose failure isolates the source and destination nodes: those SRLGs are replaced with a smaller non-isolating SRLG according to the failure model.

[3] E.g. Cisco Carrier Routing System, Juniper NorthStar SDN Controller.

While the first reason can be directly translated to cost, the second seems to be easily bypassed. Therefore, most of the related research concentrates on enabling networks to handle a large number of SRLGs by dealing with arbitrary multi-link failures [19]–[21].

Unfortunately, solutions that increase the number of SRLGs are struggling with complexity and management issues; thus they have not been deployed so far among commercially available network equipment. We believe the second criterion is more important than the first, and we should focus on keeping the number of SRLGs small. Defining larger SRLGs has two adverse side effects, (a) we require more considerable distances between the backup and working resources, (b) we require to reserve more backup resources because the network is prepared to survive an outage of a larger area with a loss of a massive amount of network capacity. For (a) the problem is that we allocate protection paths with huge detours causing significant delays, which is necessary to be resilient against regional failures but we should somehow consider the topology limitations. For example, we cannot define the whole network as a single SRLG. For (b), the implemented protection scheme should find the balance between the amount of bandwidth reserved and the size of the failure. For example, after large scale failure caused by a natural disaster reducing the amount of bandwidth network-wide is more fair, than having connection outage for specific customers and 100% bandwidth for the rest.

C. Main contributions

The main contribution of the paper is to show that all possible failures caused by a circular disk with at most k nodes interior can be represented by $O((k+1)|V|)$ SRLGs in a typical backbone network topology (see Coroll. 25), and might be at most $O((k+1)|E||V|)$ in an artificial worst-case scenario (shown in Appendix I), where $|V|$ denotes the number of nodes in the network and $|E|$ the number of links. We propose a systematic approach based on computational geometric tools that can generate the list of SRLGs in $O\left(|V|^2\left(k^5+k\log(|V|k+1)+1\right)\right)$ steps on typical networks. Besides the excellent asymptotic behavior, the obtained SRLG lists are short enough for network operators to prepare their networks to be resilient to such failure situations. We believe backbone networks operated to survive SRLG failures according to our new failure model should have significantly higher reliability.

In our previous studies [22] and [23] regional failures hitting exactly $k=0$ and $k=1$ nodes have been investigated. The lists proposed for protecting these failures have a length linear in the number of network nodes and can be calculated in $O\left(|V|\log|V|\right)$ and $O\left(|V^2|\right)$ for $k=0$ and $k=1$, respectively, on typical networks. Our paper [24] provides a simpler algorithm for arbitrary k with time complexity $O\left(\left(k^2+1\right)|V^3|\right)$ in case of stronger assumptions than present paper's. Note that this is the first study containing simulation results on the proposed SRLG lists.

On the technical side, we note that however our study heavily relies on former computational geometric results on structure k-Delaunay [25], or on its dual, the k-Voronoi diagram [17], as we are interested in the maximal edge sets hit by disks hitting k nodes, our results are not trivial consequences of the previous knowledge on neither the k-Voronoi, nor the k-edge-Voronoi diagram [26], and to the best of our knowledge, no previous study addressed the problem presented in the paper[4].

The paper is organized as follows. After Sec. I of introduction, in Sec. II our model will be presented. In Sec. III a polynomial algorithm for computing the SRLG list is introduced, and both non-parametrized and parametrized upper bounds on the running time of the algorithm are given. Next, simulation results are presented in Sec. IV. We also discuss some open theoretical problems in Sec. V, and finally, Sec. VI concludes our paper.

II. Model and Assumptions

The network is modeled as an undirected connected geometric graph $\mathscr{G}=(V,\mathscr{E})$ with $n=|V|\geq 3$ nodes and $m=|\mathscr{E}|$ edges[5]. The nodes of the graph are embedded as points in the Euclidean plane, and their exact coordinates are considered to be known. In contrast to this, precise positions of edges are not known, instead we assume that for each edge e there is a *containing polygon* (or simply *polygon*) e^p in the plane in which the edge lies (see Fig. 2a). Parameter γ will be used to indicate the maximum number of sides a containing polygon e^p can have. Note that this model covers special cases when edges are considered as polygonal chains or line segments (thus at first reading, for ease, the reader may consider the edges as line segments and γ equals two).

It will be assumed that basic arithmetic functions $(+,-,\times,/,\sqrt{\ })$ 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, i.e., there are no three such points on the same line and no four on the same circle.[6]

We will often refer to circular disks simply as disks. The disk failure model will be adapted, which overestimates the area of a disaster such that all network elements that intersect the interior of a circle c are failed, and all other network elements are untouched. It is important to note that this modeling technique does not assume that the failed region has a shape of a disk, but overestimates the size of the failed region to have a tractable problem space.

Definition 1. *A circular disk failure c hits an edge e if the polygon of the edge e^p intersects the interior of disk c. Similarly node v is hit by failure c if it is in the interior of c.*

[4] Papers [27] and [28] are dealing with loosely related topics.

[5] Graph $\mathscr{G}=(V,\mathscr{E})$ is not necessarily planar.

[6] All of the results of the paper could be extended to geometric objects in non-general position, however this would complicate our arguments lowering the paper's readability, while by an insignificant perturbation of the data one can make sure that the geometric objects are in general position.

Let $\mathscr{E}_c$ (and V_c) denote the set of edges (and nodes, resp.) 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 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 beside listing failed edges has no additional information. We consider SRLGs that represent worst-case scenarios the network must be prepared for; thus, there is no SRLG which is a subset of another SRLG.

Definition 2. *Let $\mathscr{C}$ denote the set of all circular disks in the plane, and let $\mathscr{C}_k \subseteq \mathscr{C}$ denote the set of those hitting exactly k nodes from V.*

Based on the above we can define the set of failure states the network may face after a disk failure hitting exactly k nodes.

Definition 3. *Let set $F(\mathscr{C}_k)$ denote the set of edge sets which can be hit by a disk $c \in \mathscr{C}_k$, and let $M_k = M(\mathscr{C}_k)$ denote the set of maximal edge sets in $F(\mathscr{C}_k)$.*

Note that for every $l \in \{0, \dots, k-1\}$ and $f \in M_l$ there is an $f' \in M_{l+1}$ such that $f \subseteq f'$, because any disk hitting $l \leq k$ nodes could be overestimated by a disk hitting k nodes.

As mentioned before, only the maximal edge sets will be listed as SRLGs. This study aims to offer fast algorithms computing this list for various values of k, more precisely, throughout the paper we will assume $k \in \{0, n-2\}$ since if a failure hits $n-1$ nodes, there is no node pair to communicate.

III. Algorithm for Enumerating Maximal Failures

In this section, a polynomial time algorithm is presented for computing M_k. The basic idea is that determining M_k can be decomposed into several simpler tasks, as illustrated in Fig. 1. Informally, after determining the so-called k-Delaunay graph (Def. 9 in III-B), data structures *apple* (Subsec. III-C) and *seesaw* (Appendix F) are computed, and finally M_k is determined by merging lists $M_k^{u,v}$ and M_k^w resulting from querying the apples and seesaws.

A. Basic Observations

Our first observation is the following.

Claim 4. *For every $f \in M_k$ $(k \leq n-2)$ there exists a disk $c \in \mathscr{C}_k$ such that f is hit by c, and c has at least one node of V on its boundary.*

Proof: Let f be hit by a disk $c_0 \in \mathscr{C}_k$ with centre point p. Since there are nodes of V not inside c_0, c_0 can be magnified from p until its boundary reaches a node u from V. This disk c_1 is also from $\mathscr{C}_k$ and has at least one point on its boundary, still hitting f. ■

Disk c_1 described in the proof can be further magnified while keeping its center point on ray $[up$. Here we consider two cases: either there exists a node $v \in V$, which gets on the boundary while magnifying disk $c_1 \in \mathscr{C}_k$, or the open half plane h having p inside, u on the boundary, and having the normal vector $\vec{up}$ hits k nodes.

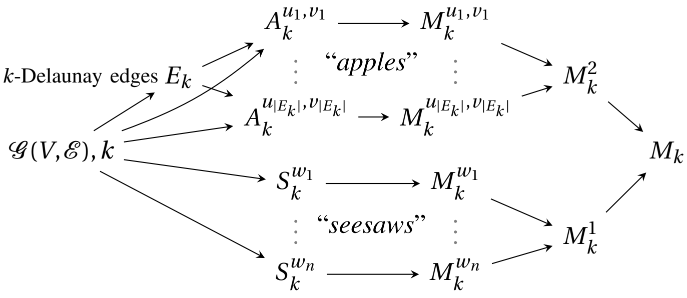


Fig. 1: Visual sketch of Algorithm 3 for determining M_k

Definition 5. *Let M_k^2 be the set of maximal failures which can be hit by a disk from $\mathscr{C}_k$ nodes having 2 nodes on its boundary. Let M_k^1 be the set of maximal failures which can be caused by a half-plane having a node on its boundary hitting exactly k nodes.*

Proposition 6. *M_k is the set maximal sets in $M_k^1 \cup M_k^2$.* ■

In the followings, we will present a way of computing M_k^2 in details using a data structure called *apple* defined in the present paper. Determining M_k^1 can be done using similar ideas, thus we present it only briefly in Appendix F-G, where the same data structure called *seesaw* is defined.

B. Graphs k-Delaunay

Definition 7. *For a node pair $u, v \in V$ let $\mathscr{C}_k^{u,v}$ denote the set of disks from $\mathscr{C}_k$ having nodes u, v on their boundary. Let $\mathscr{C}^{u,v}$ denote the set of disks from $\mathscr{C}$ having nodes u, v on their boundary. Let $M_k^{u,v} = M\left(\mathscr{C}_k^{u,v}\right)$ be the the set of failures which contain exactly the elements of M_k that can be hit by a disk $c \in \mathscr{C}_k^{u,v}$.*

Discussion after Claim 4 suggests the following simple method to compute M_k^2. First, for every node pair $\{u, v\} \subset V$, we compute a set of failures $M_k^{u,v}$.

Definition 8. *Let E_k denote the set of node-pairs $\{u, v\} \subset V$ for which $\mathscr{C}_k^{u,v} \neq \emptyset$.*

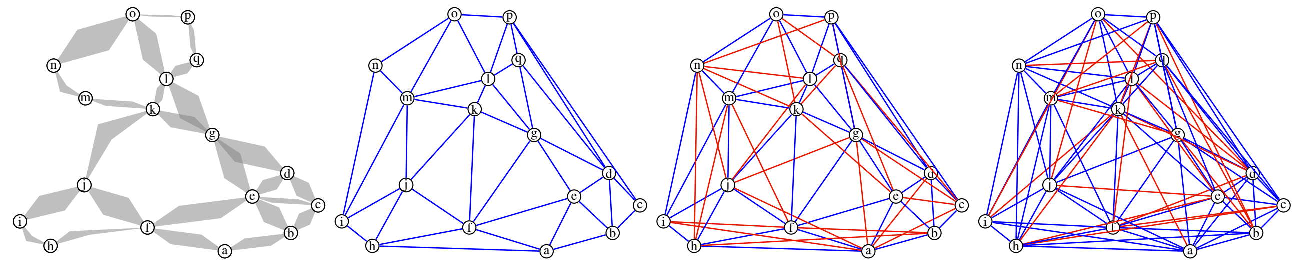
Fig. 2 shows an example of the input topology $\mathscr{G}$ and the corresponding set of node-pairs E_k for $k = 0, 1, 2$.

We can observe that by definition, M_k^2 can be computed by merging these sets $M_k^{u,v}$, formally M_k^2 is the set of maximal elements from the union of sets $M_k^{u,v}$.

Our second observation is that E_k is the edge set of the so-called k-Delaunay graph [29].

Definition 9. *Let $D_k = (V, E_k)$ denote the k-Delaunay graph induced by node set V and edge set E_k.*

The k-Delaunay graph D_k is a so-called *geometric proximity graph*. There is continuous research on a wide variety of geometric proximity graphs, where two vertices are connected



(a) Input graph $\mathcal{G} = (V, \mathcal{E})$ with containing polygons, $n = 17$, $\gamma = 4$.

(b) Graph $D_0(V, E_0)$, $|E_0| = 40$

(c) Graph $D_1(V, E_1)$, $|E_1| = 66$

(d) Graph $D_2(V, E_2)$, $|E_2| = 89$

Fig. 2: Input topology and k-Delaunay graphs D_k for $k = 0, 1, 2$.

by an edge if and only if the vertices satisfy particular geometric requirements. For example, in [30] the construction time of the k-Gabriel graph is studied, which is known to be a subgraph of D_k. In that paper, the k-Delaunay graph is determined as an intermediate step from the k-Voronoi diagram while determining the k-Gabriel graph in polynomial time. From a theorem in [25] an upper bound on $|E_k|$ can be derived. These statements are the following:

Theorem 1 (Thm. 2.4 of [30])**.** *Graph k-Delaunay $D_k = (V, E_k)$ can be constructed in $O\left(\left(k^2+1\right) n \log n\right)$.*

Theorem 2 (Thm. 2 of [25])**.** $|E_k| \leq 3(k+1)n - 3(k+1)(k+2)$.

These theorems give that for small values of k graph D_k is sparse (in other words, $\mathcal{C}_k^{u,v} = \emptyset$ for most node pairs u, v), and it can be computed fast.

C. Data Structure Apple

Let node pair $\{u, v\} \in E_k$ be given. Let us place a Cartesian coordinate system in the plane such that line uv be identical to the vertical axis y, u and v have ordinates (y coordinates) 1 and -1, respectively (see Fig. 3). Obviously, this way the centre point of any disk $c \in \mathcal{C}_k^{u,v}$ has ordinate 0.

Definition 10. *For a given node pair $\{u, v\}$, the previously described coordinate system and real number x, let $c(x)$ denote the unique disk c in $\mathcal{C}^{u,v}$, which has centre point $(x, 0)$.*

Trivially, $c()$ is a bijective function between $\mathbb{R}$ and $\mathcal{C}^{u,v}$.

Let $I_k^{u,v}$ denote the set of those numbers x, for which $c(x) \in \mathcal{C}_k^{u,v}$. If $\mathcal{C}_k^{u,v}$ is empty, then trivially $I_k^{u,v}$ is empty too. In the case when $\mathcal{C}_k^{u,v}$ is not empty, we can observe that $I_k^{u,v}$ is the union of closed intervals.

If the number of nodes in both half planes determined by line uv is not equal to k, then there exists a rightmost and a leftmost element of $\mathcal{C}_k^{u,v}$, i.e. there is a maximum x_{max} and minimum x_{min} real number in $I_k^{u,v}$, that is $x_{max} = \max I_k^{u,v}$, and $x_{min} = \min I_k^{u,v}$. Disks $c(x_{max})$ and $c(x_{min})$ have a third node w_+ and w_- on their boundary, respectively. If there are exactly k nodes on the right side of uv, then let h_+ be the right open half plane determined by line uv, and let x_{max} be sufficiently large to $\mathcal{E}_{c(x_{max})}$ to contain all the edges having polygon having a point with positive abscissa. For simplicity, sometimes $x_{max} = +\infty$ and $c(x_{max}) = h_+$ is used. Same applies for the left side of uv.

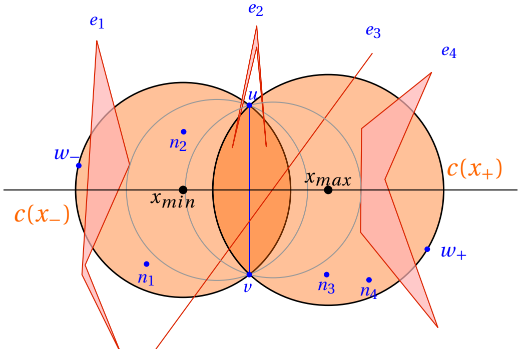


Fig. 3: Illustration of an apple with $k = 2$. Apple $A_k^{u,v}$ consists of ordered lists of nodes V_+ and V_- and ordered lists of edges $\mathcal{E}_+$ and $\mathcal{E}_-$, where $V_+ = \{n_4, n_3\}$, $V_- = \{n_2, n_1\}$, $\mathcal{E}_+ = \{e_4, e_3, e_2\}$ and $\mathcal{E}_- = \{e_3, e_2, e_1\}$. Given $\mathcal{G} = (V, \mathcal{E})$ and $\{u, v\} \subseteq V$, $A_k^{u,v}$ can be determined in $O\left(n\rho_0\gamma + k\log k + \rho_k \log \rho_k\right)$ (proof of Lemma 30). By *querying* $A_k^{u,v}$, $M_k^{u,v}$ can be computed in $O\left(\rho_k^3\right)$ (proof of Lemma 32).

Let $\mathcal{E}_{c(x_{max})}$ and $\mathcal{E}_{c(x_{min})}$ denote the edge sets hit by $c(x_{max})$ and $c(x_{min})$, respectively. To compute $M_k^{u,v}$ we use the following observation.

Claim 11. *For all $f \in F(\mathcal{C}_k^{u,v})$, $f \subseteq \mathcal{E}_{c(x_{max})} \cup \mathcal{E}_{c(x_{min})}$.*

Proof: It is easy to see that for every disk $c \in \mathcal{C}_k^{u,v}$, $c \subseteq c(x_{max}) \cup c(x_{min})$. ■

According to Claim 11, a first step towards computing $M_k^{u,v}$ is to determine the edge sets hit by $c(x_{max})$ and $c(x_{min})$. Trivially, this can be done in $O\left(m\gamma\right)$. The remaining question is how to calculate $M_k^{u,v}$ from $\mathcal{E}_{c(x_{max})} \cup \mathcal{E}_{c(x_{min})}$. Some additional notations and definitions precede the presentation of the solution.

Let ◗ denote the right side of disk $c(x_{max})$ cut by the vertical line uv, and let ◖ denote the left side of disk $c(x_{min})$ cut by the vertical line uv. For each edge $e \in \mathcal{E}_{c(x_{max})} \cup \mathcal{E}_{c(x_{min})}$ we will compute two disks: the leftmost disk which hits $e^p \cap$ ◗ and the rightmost disk which hits $e^p \cap$ ◖, which have centre points $x_+(e)$ and $x_-(e)$ respectively.

Let $\mathcal{E}_+$ denote the list of edges hit by ◗, and similarly, let

TABLE I: Table of symbols

Notation	Meaning
	General
$\mathscr{G}(V,\mathscr{E})$	the network modeled as an undirected connected geometric graph
n, m	the number of nodes $\|V\| \geq 3$ and edges $\|\mathscr{E}\|$, respectively
e^p	the containing polygon of edge e (see Fig. 2a)
V_c, $\mathscr{E}_c$	set of nodes and edges, resp., hit by a disk c
$\mathscr{C}$	the set of all circular disks in the plane
$\mathscr{C}_k$	the set of circular disks in the plane hitting exactly k nodes
$F(\mathscr{C}_k)$	the set of edge sets which can be hit by a disk $c \in \mathscr{C}_k$
M_k	the set of maximal edge sets in $F(\mathscr{C}_k)$
M_k^2	the set of maximal failures which can be hit by a disk from $\mathscr{C}_k$ having 2 nodes on its boundary
M_k^1	the set of maximal failures which can be caused by a half-plane having a node on its boundary hitting exactly k nodes
$\mathscr{C}_k^{u,v}$	the set of disks from $\mathscr{C}_k$ having nodes u,v on their boundary
$\mathscr{C}^{u,v}$	the set of disks from $\mathscr{C}$ having nodes u,v on their boundary
$M_k^{u,v}$	the the set of failures which contain exactly the elements of M_k that can be hit by a disk $c \in \mathscr{C}_k^{u,v}$
	Parameter
k	we are interested in circular disk shaped disasters hitting k nodes
γ	the maximum number of sides a containing polygon e^p can have
ρ_k	the maximum number of edges hit by a disk hitting k nodes.
	Apple
E_k	the set of node-pairs $\{u,v\} \subset V$ for which $\mathscr{C}_k^{u,v} \neq \emptyset$
$D_k(V,E_k)$	the k-Delaunay graph induced by node set V and edge set E_k
$c(x)$, h_+, h_-	For $u,v \in V$, a Cartesian coordinate system is placed in the plane such that line uv be identical to the vertical axis y, u and v have ordinates (y coordinates) 1 and -1, respectively (see Fig. 3). In this coordinate system, h_+ and h_+ are the right and left open half plane determined by line uv, respectively, and $c(x) = c(x,u,v)$ denotes the unique disk c in $\mathscr{C}^{u,v}$, which has centre point $(x,0)$, and $c(+\infty) := h_+$ and $c(-\infty) := h_-$, respectively.
$I_k^{u,v}$	the set of those numbers x, for which $c(x) \in \mathscr{C}_k^{u,v}$
x_{max}	the maximum of $I_k^{u,v}$, if exists, else if $I_k^{u,v} \neq \emptyset$, $x_{max} = +\infty$
x_{min}	the minimum of $I_k^{u,v}$, if exists, else if $I_k^{u,v} \neq \emptyset$, $x_{min} = -\infty$
$\mathscr{E}_{c(x_{max})}$, $\mathscr{E}_{c(x_{min})}$	the edge sets hit by $c(x_{max})$ and $c(x_{min})$, respectively
◗	for $u,v \in V$, the right side of $c(x_{max})$ cut by the vertical line uv
◖	for $u,v \in V$, the left side of $c(x_{min})$ cut by the vertical line uv
$x_+(e)$	for edge e, the leftmost disk which hits $e^p \cap$ ◗
$x_-(e)$	for edge e, the rightmost disk which hits $e^p \cap$ ◖
$x_+(v)$, $x_-(v)$	For nodes $w_+ \in h_+$ and $w_- \in h_-$, let $x_+(w_+)$ and $x_-(w_-)$ denote the abcissa of the centre point of circle going through u, v and w_+ or w_-, respectively.
$\mathscr{E}_+$	the list of edges hit by ◗ ordered descending by the x_+ values
$\mathscr{E}_-$	the list of edges hit by ◖ ordered descending by the x_- values
V_+	the list of nodes hit by ◗ ordered descending by the x_+ values
V_-	the list of nodes hit by ◖ ordered descending by the x_- values
V'_+	the list of nodes w in h_+ ordered decreasingly by the abscissa $x_+(z)$ of their leftmost hitting circles (going through u,v,z)
V'_-	the list of nodes z in h_- ordered decreasingly by the abscissa $x_-(z)$ of their rightmost hitting disk (going through u,v,z)
$A_k^{u,v}$	For an edge $\{u,v\} \in E_k$, apple $A_k^{u,v}$ is an ordered system $A_k^{u,v} = (V_+, V_- \mathscr{E}_+, \mathscr{E}_-)$. For each element of each list its appropriate $x_+()$ or $x_-()$ value is also stored.
A_k	the set of apples $A_k^{u,v}$
	Seesaw *(only in Appendix F)*
For data structure Seesaw for determining M_k^1, please check Appendix F.	

$\mathscr{E}_-$ be the list of edges hit by ◖ . Thus, we have $\mathscr{E}_+ \subseteq \mathscr{E}_{c(x_{max})}$ and $\mathscr{E}_- \subseteq \mathscr{E}_{c(x_{min})}$, and also $\mathscr{E}_+ \cup \mathscr{E}_- = \mathscr{E}_{c(x_{max})} \cup \mathscr{E}_{c(x_{min})}$.

Let $\mathscr{E}_+$ and $\mathscr{E}_-$ be ordered descending by the x_+ and x_- values of their elements, respectively.

Note that according to Claim 39 from the Appendix, both $x_+(e)$ and $x_-(e)$ can be computed in $O(\gamma)$.

For nodes $w_+ \in h_+$ and $w_- \in h_-$, let $x_+(w_+)$ and $x_-(w_-)$ denote the abscissa of the centre point of circle going through u, v and w_+ or w_-, respectively. We introduce V_+ and V_- similarly to $\mathscr{E}_+$ and $\mathscr{E}_-$, but instead of edges we store nodes hit by ◗ ordered descending by their x_+ values, while in V_- nodes in ◖ are stored ordered also descending, but by their x_- values. Trivially, for a node $v \in V$ both $x_+(v)$ and $x_-(v)$ can be determined in $O(1)$.

Note that while every node $v \in V$ is part of at most one of lists V_+ and V_-, edges can be part of both $\mathscr{E}_+$ and $\mathscr{E}_-$.

Now we can define the data structure *apple* for each edge of the k-Delaunay graph.

Definition 12. *For an edge $\{u,v\} \in E_k$, apple $A_k^{u,v}$ is an ordered system $A_k^{u,v} = (V_+, V_- \mathscr{E}_+, \mathscr{E}_-)$, where its composing lists are as described in the subsection before. For each element l of each list we also store its appropriate $x_+(l)$ or $x_-(l)$ value.*

D. Concept of Sweep Disk Algorithms

1) Concept: In this subsection, we highlight the paradigm of sweep disk algorithms, which is similar to the algorithmic paradigm of sweep line (sweep surface) algorithms in computational geometry.

In the case of sweep line algorithms, it is imagined that a line is moved across the plane, keeping its orientation and stopping at some event points. Geometric operations are restricted to the immediate vicinity of the sweep line whenever it ends, and the complete solution is available once the line has passed over all objects. For example, Fortune's algorithm for computing the Voronoi diagram of a point set is a sweep line algorithm [17].

Our sweep disk algorithms will scan through disk sets $\mathscr{C}^{u,v}$. In this sense in contrast to the sweep surface paradigm, our disks have different diameters, and instead of keeping orientation the invariant will be that all disks have u and v on the boundary. Thus our disk to sweep is "elastic," in the sense that it can change its diameter, but not its shape.

2) Example: Our first sweep disk algorithm is used for determining x_{max} and x_{min} for a given $A_k^{u,v}$. The algorithm works as follows. Starting from a disk $c(x) \in \mathscr{C}^{u,v}$ having centre point with abscissa $x = +\infty$ (or sufficiently large), c is swept throughout the elements of $\mathscr{C}^{u,v}$ until $x = -\infty$ (or sufficiently small). Meanwhile the number of nodes hit is followed. Numbers x_{max} and x_{min} can be determined at the first and last state when c hits exactly k nodes, respectively. (Non-existence of such moments would mean that $\{u,v\} \notin E_k$.)

Technically this can be done as follows. Let $V'_+ \subseteq V$ be the list of nodes w from h_+ ordered decreasingly by the abscissa $x_+(z)$ of their leftmost hitting circles (going through u,v,z). Similarly, let V'_- be the list of nodes z in h_- ordered decreasingly by the abscissa $x_-(z)$ of their rightmost hitting disk (going through u,v,z). Applying the fact that a node pair $z_+ \in V'_+$ and $z_- \in V'_-$ can be hit by the same disk $c \in \mathscr{C}^{u,v}$ iff $x_+(z_+) \geq x_-(z_-)$, sweeping can be imitated as in Algorithm 1. Note that for every node z in V'_+ or V'_-, $x_+(z)$ or $x_-(z)$ is stored as part of function x_+ or x_-.

From the following Proposition 13, one can check that the number of hit nodes can be easily followed with the help of an additional variable.

Algorithm 1: Determining x_{max} and x_{min} while sweeping through $\mathscr{C}^{u,v}$

```
Input: $V$ and $u, v \in V$
Output: $x_{max}$ and $x_{min}$
begin
    Compute ordered lists $V'_+$ and $V'_-$
    Merge $V'_+$ and $V'_-$ into descending ordered list $V'_\pm$ using values $x_+$
      for $V_+$ and $x_-$ for $V_-$
    $n_+, n_- \leftarrow 0$
    for $l \in \{1, \dots, |V'_\pm|\}$ do
        if $V'_\pm[l] \in V'_+$ then $n_+ += 1$
        else $n_- += 1$
        $\#_l := |V'_+| - n_+ + n_-$ // # currently hit nodes
    if $|V'_+| = k$ then $z_+ = v_\emptyset$; $x_{max} = +\infty$
    else $w_+ := V'_\pm[\min l : \#_l = k]$; $x_{max} := x_+(w_+)$
    if $|V'_-| = k$ then $z_- = v_\emptyset$; $x_{min} = -\infty$
    else $w_- := V'_\pm[\max l : \#_l = k]$; $x_{min} := x_-(w_-)$
    return $x_{max}$ and $x_{min}$
```

Proposition 13. *Let $c \in \mathscr{C}^{u,v}$. If $V'_+[i-1]$ is not hit by c, then all the preceding elements in V'_+ are not hit by c. If $V'_+[i]$ is hit by c, then all the following elements in V'_+ are hit by c.*

Similarly, if $V'_-[i-1]$ is hit by c, then all the preceding elements are hit by c. If $V'_-[i]$ is not hit by c, then all the following elements are not hit by c. ■

Claim 14. *For a given edge $\{u, v\} \in E_k$, both x_{max} and x_{min} can be determined in $O(n \log n)$ time.*

Proof: According to those written in this subsection, both V'_+ and V'_- can be determined in $O(n \log n)$ the dominant step being a sorting algorithm. Sweeping can be trivially done in $O(n)$; meanwhile, both x_{max} and x_{min} can be determined. ■

Proposition 15. *Both V_+ and V_- can be determined in $O(n \log n)$.* ■

E. Determining Apples

Claim 16. *For a given $\{u, v\} \in E_k$, apple $A_k^{u,v}$ can be determined in $O(m(\log m + \gamma))$.*

Proof: If $\{u, v\} \in E_k$, then x_{max}, x_{min}, V_+ and V_- can be determined in $O(n \log n)$ according to Claim 14 and Prop. 15, thus it remains to determine $\mathscr{E}_+$ and $\mathscr{E}_-$. With this aim it is enough to compute the $x_+(e)$ and $x_-(e)$ values for every edge, then collect in $\mathscr{E}_+$ those edges e for which $x_+(e) \le x_{max}$ and similarly in $\mathscr{E}_-$ those edges e for which $x_-(e) \ge x_{min}$. Finally, edges in $\mathscr{E}_+$ and $\mathscr{E}_-$ have to be sorted descending according to their x_+ and x_- values, respectively. m polygons of edges (each having at most γ sides) have to be checked and sorted which gives a total complexity of $O(m(\gamma + \log m))$. ■

Definition 17. *Let A_k be the set of apples $A_k^{u,v}$.*

Corollary 18. *For a given k, knowing E_k, the set of apples A_k can be determined in $O((k+1)\, nm(\gamma + \log m))$.*

Proof: Since by Thm. 2 $|E_k| < 3(k+1)\, n$, we deduct that $O((k+1)\, n)$ apples have to be determined. According to Claim 14, an apple can be constructed in $O(m(\log m + \gamma))$, which completes the proof. ■

Algorithm 2: Querying an apple

```
Input: Apple $A_k^{u,v}$ Output: Set $M_k^{u,v}$ of locally maximal failures.
begin
    Merge $V_+$, $V_-$, $\mathscr{E}_+$ and $\mathscr{E}_+$ into descending ordered list $G$ using
      values $x_+$ for $V_+$ and $\mathscr{E}_+$ and $x_-$ for $V_-$ and $\mathscr{E}_-$;
    $n_+, n_-, e_+, e_- \leftarrow 0$;
    for $l \in \{1, \dots, |G|\}$ do
        if $l \in V_+$ then $n_+ += 1$;
        if $l \in V_-$ then $n_- += 1$;
        if $l \in \mathscr{E}_+$ then $e_+ += 1$;
        if $l \in \mathscr{E}_-$ then $e_- += 1$;
        $\#_{n,l} := |V_+| - n_+ + n_-$ // # curr. hit nodes
        $\#_{e,l} := |\mathscr{E}_+| - e_+ + e_-$ // # curr. hit edges
        $e_{+,l} := e_+$ // $\mathscr{E}_+[i]$ is hit iff $i \ge e_{+,l}$
        $e_{-,l} := e_-$ // $\mathscr{E}_-[i]$ is hit iff $i \le e_{-,l}$
    Det. $L$, the set of indexes $l$, for which $\#_{n,l} = k$;
    Det. $I_e$, the sequence of numbers $\#_{e,l} : l \in L$;
    Det. $M_e$, the set of indexes $l$ of local maximums of $I_e$;
    $(M_k^{u,v})'$ := hit edge sets in $l \in M_e$ disk positions;
    // Can be det. using $\mathscr{E}_+, \mathscr{E}_-, e_{+,l}, e_{-,l}$
    $M_k^{u,v} \leftarrow$ maximal elements of $(M_k^{u,v})'$;
    return $M_k^{u,v}$
```

F. Computing the Set of SRLGs by Sweeping Through Each Apple

Claim 19. *Let $e \in \mathscr{E}_+$, and $f \in \mathscr{E}_-$. They can be hit by the same $c \in \mathscr{C}_k^{u,v}$ if $x_+(e) \le x_-(f)$ or $x_+(f) \le x_-(e)$.*

Proof: An edge e can be hit by circle $c(x)$ if $x_+(e) \le x$ or $x \le x_-(e)$. ■

Determining $M_k^{u,v}$ from apple $A_k^{u,v}$ can be done with the help of a sweep disk algorithm as a subroutine of Algorithm 2 similar to Algorithm 1, the only difference is that here we have to check both the set of currently hit edges and the number of currently hit nodes at the same time.

On one hand, while sweeping through $\mathscr{C}^{u,v}$ with $c(x)$ (while x decreasing), nodes are also getting hit or not hit by the actual $c(x)$, thus it is not necessarily permanently part of $\mathscr{C}_k^{u,v}$ during the sweep disk algorithm. On the other hand, any edge e having e^p intersecting segment $[u, v]$ or for which $x_+(e) \le x_-(e)$ should be stored exactly once in any element of $M_k^{u,v}$.

Claim 20. *Querying $A_k^{u,v}$, Algorithm 2 calculates $M_k^{u,v}$ in $O(m^3)$.*

Proof: Correctness of the algorithm can be easily checked. Since while sweeping an edge can get hit or unhit at most once on one side of line uv, there are at most $O(m)$ failures with locally maximal cardinalities, each of them having $O(m)$ edges, thus $(M_k^{u,v})'$ has $O(m)$ elements of $O(m)$ size. Trivially, the number of currently hitting nodes can be monitored in $O(n)$ total time as in Alg. 1. Since we have an ordering of the edges, every pair of sets from $(M_k^{u,v})'$ can be compared in $O(m)$. This means that from $(M_k^{u,v})'$, $M_k^{u,v}$ can be determined in $O(m^3)$. It can be checked that all the other operations have complexity at most $O(m^3)$. ■

Corollary 21. *Known E_k, lists $M_k^{u,v}$ for all $\{u, v\} \in E_k$, can be determined in $O((k+1)\, nm^3)$.* ■

Algorithm 3: Algorithm for computing M_k with a table on time complexities. (Refer to Table I for notations.)

```
   Input: $\mathscr{G}(V,\mathscr{E})$, $k$ Output: $M_k$ begin
       Determining $M_k^2$;
1          Determine $E_k$;
2          Determine set $A_k$ of nonempty apples;
3          Query apples from $A_k$;
4          Merge lists $M_k^{u,v}$ into $M_k^2$;
       Determining $M_k^1$ // See App. F-G for details
5          Determine set $S_k$ of nonempty seesaws;
6          Query seesaws from $S_k$;
7          Merge lists $M_k^w$ into $M_k^1$;
8      Merge lists $M_k^2$ and $M_k^1$ into $M_k$;
9      return $M_k$
```

Complexity	Non-parametrized	Parametrized
Step 1	$O((k^2+1)n\log n)$	$O((k^2+1)n\log n)$
Step 2	$O(n(k+1)m(\log m+\gamma))$	$O\big((k+1)\,n\big(n\rho_0\gamma+k\log k+\rho_k\log\rho_k\big)\big)$
Step 3	$O(n(k+1)m^3)$	$O(n(k+1)\rho_k^3)$
Step 4	$O(n^2(k^2+1)m^3)$	$O(n^2(k^2+1)\rho_k^3)$
Step 5	$O(nm\gamma\log m)$	$O(n^2\rho_0\gamma\log(n\rho_0))$
Step 6	$O(nm^3)$	$O(n\rho_0+\rho_k^3)$
Step 7	$O(n^2m^3)$	$O(n^2\rho_k^3)$
Step 8	$O(n^2(k+1)m^3)$	$O(n^2(k+1)\rho_k^3)$
Total for $\gamma=O(1)$	$O(n^2(k^2+1)m^3)$	$O(n^2((k^2+1)\rho_k^3+\rho_0\log(n\rho_0)))$

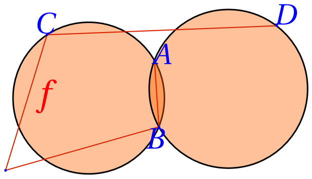


Fig. 4: In the setting above, circle c_{ABC} hits the whole link set $\mathscr{E}$, while no $c\in\mathscr{C}_0^{B,D}$ hits link f. Thus, however $M_0^{B,D}$ is not empty, it does not contain any elements of M_0.

Here we use the assumption that the corner points of the polygons of the edges are in general position.

G. Algorithm for Computing Maximal Failures

As presented before M_k^2 can be calculated by determining and querying the apples, and finally merging the obtained lists $M_k^{u,v}$ in M_k^2. Note that there is a valid need of comparing lists of locally maximal failures (see Fig. 4). M_k^1 can be computed very similarly to M_k^2 (as shown in Appendix F-G). Finally, in order to get M_k, M_k^2 and M_k^1 have to be merged. This way the scheme of our algorithm could be written as in Alg. 3.

Complexity bounds on the non-parametrized computing time and length of M_k are summarized as part of the table in Alg. 3. Although in Appendix I it is shown that there are some artificial networks where these asymptotic bounds are relatively good estimations, we would like to focus on the running time and output size on the real networks, which are nearly planar. Thus, after introducing a new parameter, we present parametrized bounds proven in Appendix B-G.

Intuitively, a $c\in\mathscr{C}_k$ cannot hit too many edges. Thus we introduce graph density parameter ρ_k, which describes this phenomenon.

Definition 22. *For all $i\in\{0,n-2\}$, let ρ_i be the maximum number of edges hit by a disk from $\mathscr{C}_i$.*

The parametrized bounds are the following:

Lemma 23. *M_k^2 can be computed in $O\big(n^2\big((k^2+1)\,\rho_k^3+(k+1)\,(\rho_0\gamma)\big)\big)$. M_k^2 has $O\big(n\,(k+1)\,\rho_k\big)$ elements with at most ρ_k edges.*

Proof of Lemma 23 can be found in Appendix E.

Besides computing M_k^2 one have to deal with computing M_k^1. When computing M_k^1, the vague idea is to give a geometric algorithm in a way similar to the sweep disk algorithm for querying the apples. Now instead of imaginary sweeping a disk, we rotate a half plane around every node $v\in V$ until it makes a total turn, and meanwhile, check for hit edge sets with locally maximal cardinalities hit by half planes hitting exactly k nodes. After this, the maximal elements of the obtained lists are collected in M_k^1. Now follows Lemma 24 for computing M_k^1. We kindly ask the reader to see Appendix F-G for its detailed proof.

Lemma 24. *M_k^1 can be constructed in $O(n^2(\rho_0\gamma\log n\rho_0+\rho_k^3))$ and has $O(n\rho_k)$ elements, each containing at most ρ_k edges.*

Theorem 3. *M_k can be computed in $O\big(n^2\big((k^2+1)\,\rho_k^3+\rho_k\gamma+\big(k+1+\log(n\rho_0)\big)\,\rho_0\gamma\big)\big)$. M_k has $O\big(n\,(k+1)\,\rho_k\big)$ elements with at most ρ_k edges.*

Proof: Based on Lemmas 23 and 24, both M_k^2 and M_k^1 can be computed in the proposed time, have at most the proposed amount of elements containing at most ρ_k edges. The proof will be completed by showing that the merger of M_k^2 and M_k^1 can be done in $O(n^2(k+1)\rho_k^3)$, which is true because of the followings.

We only have to compare all the pairs $\{p_2,p_1\}$ made up of a $p_2\in M_k^2$ and $p_1\in M_k^1$, which means $O((n(k+1)\rho_k)(n\rho_k))$ pairs. Each comparision can be made in $O(\rho_k)$,[7] which gives the proposed complexity. ■

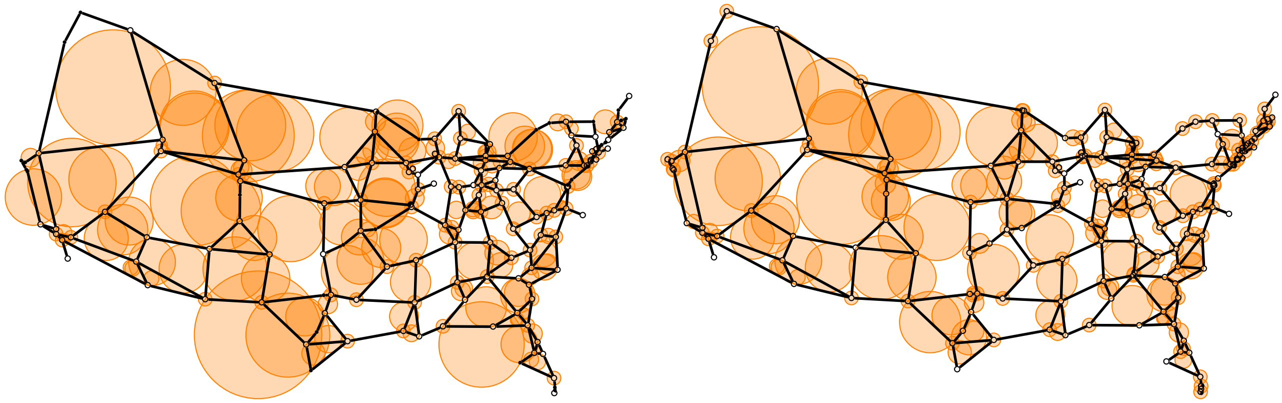
Corollary 25. *If ρ_k is $O(k+1)$, then M_k has $O(n\,(k+1))$ elements. If in addition γ is upper bounded by a constant, M_k can be computed in $O(n^2(k^5+\log(nk+1)+1))$.*[8]

IV. Simulation Results

In this section, we present numerical results that validate our model and demonstrate the use of the proposed algorithms on some realistic physical networks. The algorithms were implemented in Python version 3.5 using various libraries. No special efforts were made to make the algorithm space or time optimal. The output of the algorithm is a list of SRLGs so that

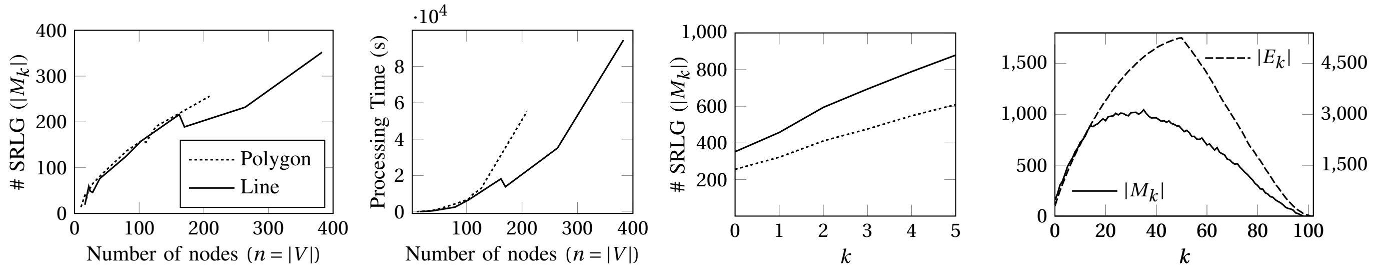
[7] Since link sets are ordered lexicographically.

[8] To be exact, $\gamma=O(\max\{1,\frac{k^5+1}{k+\log n}\})$ still yields the proposed complexity.



(a) *Polygon*: there are 190 SRLGs with average of 2.98 links and $\rho_0 = 5$. (b) *Line*: there are 216 SRLGs with average of 2.79 links and $\rho_0 = 5$.

Fig. 5: The SRLGs of $k = 0$ are visualized for the two cases (a) links are polygonal line segments, and (b) the corner points of the polygonal line segments are treated as degree two nodes and all links are straight lines. In order to have a perspicuous illustration, each SRLG is drawn with the smallest possible circular disk that covers all of its links, even if the disk has nodes interior.



(a) The increase of the number of SRLGs with respect to the number of nodes, for $k = 0$. (b) Measured time complexity. (c) The average the number of SRLGs over all networks. (d) The number of SRLGs and the number of edges in graph k-Delaunay for US Deltacom

Fig. 6: Comparison of the two interpretation of the input topologies: links are polygonal line segments, or the corner points of the polygonal line segments are treated as degree two nodes and all links are straight lines.

no SRLG contains the other. The network topologies with the obtained list of SRLGs for various k are available online[9].

First, we interpret the input topologies in two ways:

polygon where links are polygonal chains, and

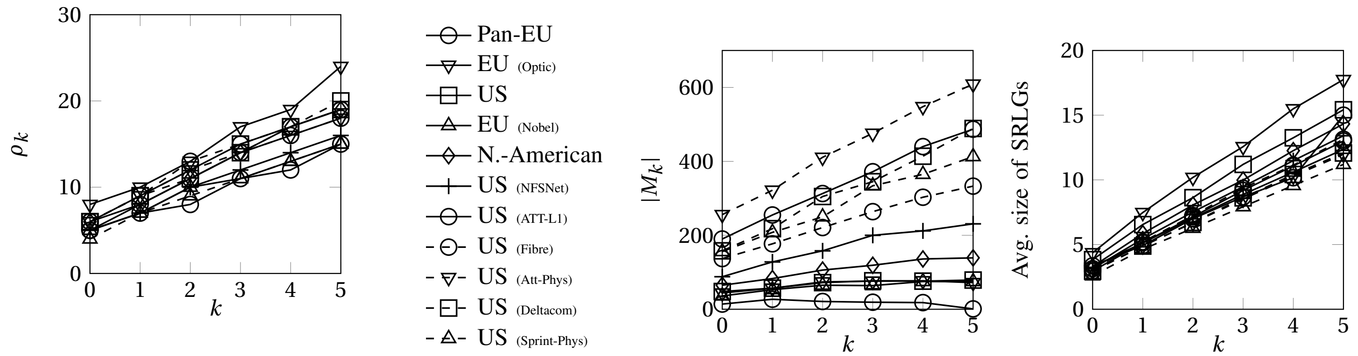
line where the corner points of the polygonal links are substituted with nodes (of degree 2). Here links are line segments.

The second interpretation is artificial, and we mainly use it for verification. Intuitively, the two interpretations result in very different results, as the latter has much more nodes in the network and thus the regional failures with k nodes interior must be smaller. Fig. 5 shows example results for both interpretations of the US ATT-L1 network. The US fiber network has 126 nodes and 208 links as polygonal chains, where the links have 36 corner nodes in total. After transforming it into a network of line segments, we will have a larger network with 162 nodes and 244 links. The transformed network has 30% more nodes; however, the number of SRLGs required for $k = 0$ is just 14% more, which is a sub-linear increase. Surprisingly, after the transformation, the SRLGs became a bit smaller (average number of links $2.98 \rightarrow 2.79$), and the variance in the size of SRLGs is increased from 0.7 to 0.83. It is because in the transformed a great number of very small SRLGs appeared having the two adjacent links for most of the degree 2 nodes.

TABLE II: Results of physical backbone topologies of [31].

Name	$\|V\|$		$\|\mathscr{E}\|$		# SRLG $k = 0$		# SRLG $k = 1$	
	Polygon	Line	Polygon	Line	Polygon	Line	Polygon	Line
Pan-EU	10	16	16	22	14	19	27	35
EU (Optic)	17	22	40	45	44	59	57	71
EU (Nobel)	19	28	32	41	36	46	53	81
US [7]	21	24	39	42	48	49	57	64
N.-American	28	39	50	61	65	76	83	97
US (NFSNet)	44	79	73	108	88	124	128	172
US (Fibre)	81	170	141	230	137	189	177	249
US (Deltacom)	103	103	302	302	158	158	218	218
US (Sprint-Phys)	111	264	160	313	156	232	208	307
US (ATT-L1)	126	162	208	244	190	216	255	285
US (Att-Phys)	209	383	314	488	256	352	322	457

[9] https://github.com/jtapolcai/regional-srlg



(a) The edge density (ρ_k)

(b) The number of SRLGs ($|M_k|$).

(c) The average size of SRLGs

Fig. 7: The edge density, number and size of SRLGs for each network and $k = \{0, \ldots, 5\}$ in case of polygonal chain links.

A. The list of SRLGs in practice

Table II shows a comparison for $k = 0$ and $k = 1$ among eleven physical backbone topologies taken from [31]. The columns are: network name, the number of nodes, the number of links, the number of SRLGs, all for both cases where links are polygonal line segments or each corner points of the polygonal links are substituted with a degree 2 node.

The 126-node US (ATT-L1) network was covered with 190 SRLGs, which is less than listing every single node and link as an SRLG. Fig. 5 shows these SRLGs, intuitively each corresponds to a mid-size regional failure. The SRLGs meet our intuition that there are more network nodes in the crowded areas, and thus it generates more SRLGs for them, while in the less crowded areas are covered with SRLGs corresponding to bigger areas. In practice, it is important to have small SRLGs because it strongly influences the performance of the survivable routing algorithms. On Fig. 5 the SRLGs are relatively small, each SRLG contains a bit less than 3 links on average. Fig. 7c shows how the average size of SRLGs with respect to k for all networks. It has a slightly sub-linear increase with k. Note that, the length of the list of SRLGs never exceeded 6000 items in any networks and parameter settings examined.

Next, we have evaluated what would be the radius of the circular disk with $k = 0, 1, 2, 3$ nodes when we know the GPS positions of the nodes (in case of network US ATT-L1). We have performed a Monte Carlo simulation where we pick random locations and compute the maximum radius with $k = 0, 1, 2, 3$ nodes, which is the distance of the closest, the second, third, and fourth closest nodes. Fig. 8 shows the cumulative distribution function of the actual radius of the circular disk failures. For example, if we cut the smallest and largest 20% the SRLGs generated for $k = 0$ nodes corresponds to diameter 80-200km.

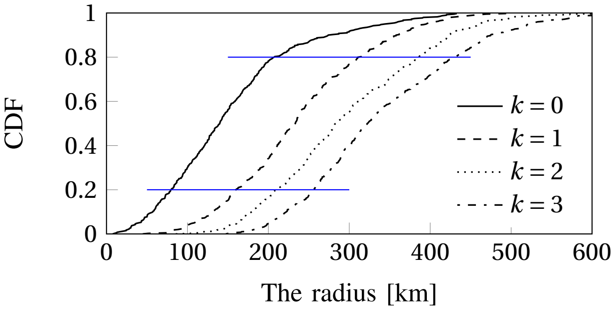


Fig. 8: The cumulative distribution function of the radius of the disk having a given k number of nodes interior.

B. Tightness of Corollary 25

In this subsection we compare the presented parameterized worst case analysis with the obtained simulation results. Corollary 25 assumes that ρ_k increases linearly with $(k+1)$ for all networks. Fig. 7a shows that the edge density increases linearly with $(k+1)$ as we expected. In Corollary 25 a near quadratic worst case increase of computation time was showed with respect to the network size n for fixed k (more precisely it is $O(n^2 \log n)$). It is well reflected in Fig. 6b. Note that we measured the runtime of the algorithms on a commodity computer with CPU at 2.4 GHz and 32 GB of RAM without optimizing the implementation for speed. Corollary 25 provides a linear upper bound on the number of SRLGs with respect to the network size n for fixed k. It is also reflected in Fig. 6a which is a graphical illustration for $k = 0$, where we may even have the intuition of a sub-linear growth. Based on Table II the slope of the curve can be estimated as the number of SRLGs is roughly $\approx 1.2n$ for $k = 0$, and $\approx 2.2n$ for $k = 1$. Corollary 25 provides a linear upper bound on the number of SRLGs with respect to k if the network (n) is fixed. This is illustrated in Fig. 6c where the average number of SRLG is shown for all networks for small k. Here we can experience a slightly sub-linear increase. Fig. 7b shows the increase in the number of SRLGs for each network independently for the same range of k. We experienced sub-linearity for all networks, which we further discuss later. Overall, numerical evaluation supports the parameter selection used in the parametrized complexity analysis. We conjecture that the Corollary 25 is close to the experienced performance, and there is little hope for further improving it analytically.

To reveal the reasons why the number of SRLGs increases sub-linearly with respect to k, Fig. 6d shows the function graphs of both $|M_k|$ and $\frac{|E_k|}{3}$ for all k values on Network US Deltacom, respectively. As we can see, for $k \le 15$, $|M_k| \approx \frac{|E_k|}{3}$, while for bigger k values $|M_k| < \frac{|E_k|}{3}$. We can deduct that $|M_k| \le \frac{|E_k|}{3}$ for every $k \in \{0, \dots, n-2\}$. In other words, the average number of SRLG-s per apple is $\le \frac{1}{3}$. By Thm. 2, this also means that $|M_k|$ is $O((k+1)(n-k))$, wich induces a sub-linear growth of $|M_k|$ with respect to k.

V. Open problems

The presented algorithm computes M_k in $O\big(n^2\big(k^5 + k\log(nk+1)+1\big)\big)$ (if γ is constant and ρ_k is $O(k+1)$). A natural question is whether it is possible to compute M_k quicker. For $k=0$ the answer is yes in case of usual networks: as presented in [22], M_0 can be determined in $O(n(\log n + \rho_0^3 \tau_0))$, where τ_0 is an additional parameter depending on local properties of the embedding of the network in the plane (and edges are considered as line segments). In other words, there exists an algorithm to compute M_0 in $f(\rho_0, \tau_0)O(n \log n)$.

We believe that there exist similar algorithms for determining M_k in $f\big(k, \rho_k, \gamma, \dots\big)\, O\big(n \log n\big)$, where f depends only on k and on 'local' properties of the embedding. This near-linear complexity in n would allow M_k to be computed even quicker in case of huge networks too. Unfortunately, presenting this kind of algorithms would exceed the limits of this paper; thus it will be part of future work.

VI. Conclusions

In this paper, we propose a fast and systematic approach to enumerate the list of possible regional failures the network should be prepared for. We define regions as circular disks with maximum size such that the number of nodes inside is at most k. The proposed failure model gives a reasonable conservative description of the possible regional failures with a minor drawback of overestimating the size of the failed region, but with the following three benefits that

- It can be computed efficiently, even for very large networks.
- The list of SRLGs is short enough to fit in the current router configurations.
- A schematic topology map discovered by automatic tools is sufficient to compute the list of SRLGs, and does not require precise knowledge on the geometry of the network, i.e. the exact physical location of the nodes and links, the physical distances in the network topology, etc.

Using the proposed algorithm, operators can protect their networks against regional failures by running the current self-healing mechanisms with the obtained list of SRLGs.

References

[1] J. P. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, no. 8, pp. 1245–1265, 2010.

[2] J. Rak, "Resilient routing in communication networks," in *Computer Communications and Networks*, 2015.

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

[4] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *Communications Magazine, IEEE*, vol. 50, no. 2, pp. s12–s20, 2012.

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

[6] J. Heidemann, L. Quan, and Y. Pradkin, *A preliminary analysis of network outages during hurricane Sandy*. University of Southern California, Information Sciences Institute, 2012.

[7] F. Dikbiyik, M. Tornatore, and B. Mukherjee, "Minimizing the risk from disaster failures in optical backbone networks," *Journal of Lightwave Technology*, vol. 32, no. 18, pp. 3175–3183, 2014.

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

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

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

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

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

[13] 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 Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1525–1538, 2013.

[14] S. Trajanovski, F. Kuipers, P. Van Mieghem *et al.*, "Finding critical regions in a network," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2013, pp. 223–228.

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

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

[17] F. Aurenhammer, "Voronoi diagrams: a survey of a fundamental geometric data structure," *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.

[18] S. Neumayer, A. Efrat, and E. Modiano, "Geographic max-flow and min-cut under a circular disk failure model," *Computer Networks*, vol. 77, pp. 117–127, 2015.

[19] S. Ramasubramanian and A. Chandak, "Dual-link failure resiliency through backup link mutual exclusion," *IEEE/ACM Trans. Networking*, vol. 16, no. 1, pp. 157–169, 2008.

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

[21] T. Elhourani, A. Gopalan, and S. Ramasubramanian, "IP fast rerouting for multi-link failures," *IEEE/ACM Transactions on Networking (ToN)*, vol. 24, no. 5, pp. 3014–3025, 2016.

[22] B. Vass, E. Bérczi-Kovács, and J. Tapolcai, "Shared risk link group enumeration of node excluding disaster failures," in *Int. Conference on Networking and Network Applications (NaNA)*, 2016.

[23] B. Vass, E. Bérczi-Kovács, and J. Tapolcai, "Enumerating circular disk failures covering a single node," in *Int. Workshop on Resilient Networks Design and Modeling (RNDM)*, Halmstad, Sweden, Sep. 2016.

[24] ——, "Enumerating shared risk link groups of circular disk failures hitting k nodes," in *Proc. International Workshop on Design Of Reliable Communication Networks (DRCN)*, Munich, Germany, march 2017.

[25] M. Abellanas, P. Bose, F. H. J. García, C. M. Nicolás, and P. A. Ramos, "On structural and graph theoretic properties of higher order Delaunay graphs," in *International Journal of Computational Geometry & Applications, Volume 19, Issue 06*, 2009.

[26] E. Papadopoulou and M. Zavershynskyi, "The higher-order voronoi diagram of line segments," *Algorithmica*, vol. 74, no. 1, pp. 415–439, Jan 2016. [Online]. Available: https://doi.org/10.1007/s00453-014-9950-0

[27] R. Pinchasi and G. Rote, "On the maximum size of an anti-chain of linearly separable sets and convex pseudo-discs1," 2008. [Online]. Available: https://arxiv.org/pdf/0707.0311.pdf

[28] T. M. Chan, "On levels in arrangements of surfaces in three dimensions," 2012. [Online]. Available: http://tmc.web.engr.illinois.edu/surf.pdf

[29] F. Aurenhammer, R. Klein, and D. Lee, *Voronoi diagrams and Delaunay triangulations*. World Scientific Publishing Co Inc, 2013.

[30] T.-H. Su and R.-C. Chang, "The *k*-Gabriel graphs and their applications," in *Algorithms. Lecture Notes in Computer Science, vol.450.*, Berlin, Heidelberg, 1990.

[31] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "Sndlib 1.0: survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[32] [Online]. Available: http://www.cs.bsu.edu/homepages/fischer/math345/stereo.pdf

Appendix

A. Protecting regional link 0-node failures ensures node disjointness

Claim 26. *Let P and B be an SRLG-disjoint primary and backup paths according to regional link 0-node failures. The paths P and B are node disjoint, apart from the terminal nodes.*

Proof: Assume indirectly that P and B have a common node v in their interior. Let us pick two edges $e_p \in P$ and $e_b \in B$ of G that are adjacent of v. Let p denote the closest corner point of e_p with node v if there is any, otherwise its endpoint on P. Similarly we define b for e_b. Note that points v, b and p are not on the same line according to our assumptions in Sec. II; thus, the is a circular disk of any size that covers both e_p and e_b but not node v. We can select a small enough radius for this circular disk which covers both e_p and e_b and does not have any other nodes interior. The proof follows. ■

Parametrized Algorithm for Enumerating Maximal Failures in M_k^2

B. Parametrized Complexity Bounds for Determining Apples

Up to this point the fact that $\mathscr{G}(V,\mathscr{E})$ is, in fact, a graph of a communication network, and thus it is 'almost planar' was not used. Intuitively, an almost planar graph has $O(n)$ edges. Parameters ρ_i ($i \in \{0,\dots,n-2\}$) denote the maximum number of edges hit by a disk from $\mathscr{C}_i$ (Def. 22), thus they help in formalizing this intuition.

Since parameters ρ_i measure local properties of the networks, often it will be assumed that these parameters are not exceeding a constant. For example, ρ_0 is not going to be large since where there are many edges, it is likely to appear a node too.

Observation 27. *For any $0 \le i < j \le n-2$, $\rho_i \le \rho_j$.* ■

Claim 28. *In any apple $A_k^{u,v} \in A_k$, $|\mathscr{E}_+| \le \rho_k$ and $|\mathscr{E}_-| \le \rho_k$.*

Proof: All edges in $\mathscr{E}_+$ are hit by $c(x_{max})$ which by definition hits at most ρ_k edges. Similar for $\mathscr{E}_-$. ■

Lemma 29. *The number of edges is $O(n\rho_0)$, more precisely $m \le (2n-5)\rho_0$.*

Proof: Consider the Delaunay triangulation D_0, which is a planar graph, and thus $|E_0| \le 3n-6$. Since every Delunay triangle has 3 Delaunay edges and a Delaunay edge is the edge of at most 2 Delaunay triangles, and there are at least 3 Delaunay edges on the convex hull of V, the number of Delaunay triangles is at most

$$\frac{2|E_0|-3}{3} \le \frac{2}{3}(3n-6)-1 = 2n-5.$$

Since the polygon of every edge in $\mathscr{E}$ intersects at least one triangle, and every triangle can be covered by a disk $c \in \mathscr{C}_0$, which intersects at most ρ_0 polygon of edges of the network, we get that the number m of edges cannot be larger than ρ_0 times the number of the Delaunay triangles. We get $m \le (2n-5)\rho_0$. ■

Lemma 30. *If E_k is given, set A_k of apples can be calculated in $O\big((k+1)\,n\,(n\rho_0\gamma + k\log k + \rho_k \log\rho_k)\big)$.*

Proof: There are $|E_k| \le 3(k+1)n$ apples to determine (Thm. 2). For each, V_+ and V_- can be determined in $O(n+k\log k)$, then (based on Lemma 29) $O(n\rho_0)$ edges have to be checked if they are in the apple, each in $O(\gamma)$ time. After this, based on Claim 28, there are $O(\rho_k)$ edges to order, which gives the proposed complexity. ■

Corollary 31. *If ρ_k and γ is upper bounded by a constant and E_k is given, then A_k can be determined in $O\big(n^2\,(k+\log n)\big)$.* ■

C. Parametrized Bound for Determining $M_k^{u,v}$ for Apple $A_k^{u,v}$

Lemma 32. *For all the apples $A_k^{u,v}$ in A_k, sets $M_k^{u,v}$ can be determined in $O\big(n\,(k+1)\,\rho_k^3\big)$.*

Proof: Based on Thm. 2, there are $|E_k| \le 3n(k+1)$ apples to query. We claim that each of them can be queried in $O(\rho_k^3)$. Knowing apple $A_k^{u,v}$, $(M_k^{u,v})'$ can be determined in $O(\rho_k^2)$, following the steps of Alg. 2. After this, $M_k^{u,v}$ can be determined by comparing each pair of elements of $(M_k^{u,v})'$ and eliminating its non-maximal and redundant members. With this purpose, $O(\rho_k^2)$ comparisons have to be made, each of them has $O(\rho_k)$ complexity. ■

D. Parametrized Complexity Bound on Merging Lists $M_k^{u,v}$

Lemma 33. *M_k^2 can be computed in $O\big(n^2\,(k^2+1)\,\rho_k^3\big)$ from lists $M_k^{u,v}$.*

Proof: By Thm. 2, there are $|E_k| \le 3n(k+1)$ lists containing $O(\rho_k)$ sets containing $O(\rho_k)$ edges. First determine an ordering on the set of edges $\mathscr{E}$, and sort all the candidate sets of edges according to this ordering, each in $O(\rho_k \log\rho_k)$ time. M_k^2 can be computed by comparing all the set pairs (and eliminating the redundant or non-maximal elements), which means $O(n^2(k^2+1)\rho_k^2)$ comparisons. Since (due to the ordering) comparing two sets takes $O(\rho_k)$ time, the total complexity is $O(n^2(k^2+1)\rho_k^3)$. ■

E. Parametrized Complexity for Computing M_k^2

Now follows the proof of Lemma 23, which states the followings. M_k can be computed in $O\left(n^2\left(\left(k^2+1\right)\rho_k^3+(k+1)\rho_0\gamma\right)\right)$. M_k has $O\left(n(k+1)\rho_k\right)$ elements with at most ρ_k edges.

Proof: As presented previously (Thm. 1 and Lemmas 30, 32 and 33), each of the corresponding four phases of Alg. 3 can be examined in the proposed complexity. There are $|E_k| \leq 3n(k+1)$ lists $M_k^{u,v}$ to merge, each of them has at most ρ_k edges, completing the proof. ■

Parametrized Algorithm for Enumerating Maximal Failures in M_k^1

In this section a sketch of an algorithm will be presented for proving Lemma 24, which states that M_k^1 can be constructed in $O\left(n^2\left(\rho_0\gamma\log\left(n\rho_0\right)+\rho_k^3\right)\right)$ and has $O\left(n\rho_k\right)$ elements, each containing $O\left(\rho_k\right)$ edges.

F. Data Structure Seesaw

For every node $w \in V$, a data structure is built containing both the direction of every node $z \in V\setminus\{w\}$ related to v and the interval of directions where the polygon of each edge can be seen from w. Nodes and edges also have to be sorted according to this information (similarly to data structure apple). Let us call this data structure *seesaw*, and let S_k^w denote the previously described seesaw. For a given k, let the set of seesaws be denoted by S_k. Let the list of locally maximal failures resulting from querying seesaw S_k^w be denoted by M_k^w.

Claim 34. *Any seesaw S_k^w can be calculated in $O\left(n\rho_0\gamma\log\left(n\rho_0\right)\right)$ and has a total length of $O\left(n\rho_0\right)$.*

Proof: Trivially, the direction of nodes from $z \in V \setminus \{w\}$ can be determined and sorted in $O(n\log n)$. Also, as by Lemma 29, the number of edges is $O(n\rho_0)$, intervals of directions corresponding to polygons of edges can be calculated and sorted both by their minimum and maximum values on $O(n\rho_0\gamma\log(n\rho_0))$. The proof follows. ■

G. Querying Seesaws

Claim 35. *M_k^w can be determined from S_k^w in $O\left(n\rho_0+\rho_k^3\right)$, and has at most $2\rho_k$ elements, each of them containing at most ρ_k edges.*

Proof: Since each set in M_k^w can be hit by a disk in $\mathscr{C}_k$ having w on the boundary, the fact that the elements of M_k^w contain at most ρ_k edges is trivial by definition.

Now we prove $|M_k^w| \leq 2\rho_k$. If there is no half plane having w on the boundary hitting exactly k nodes, then the claim is trivial, otherwise, let h_0^w be such a half plane. Let h_+^w be the unique half plane which satisfies the followings: it is the rotation of h_0^w with $\angle \leq \pi$, it covers exactly k nodes, and no other half plane covers exactly k nodes which is the rotation of h_+^w with an angle $\in [\angle, \pi]$. Let h_-^w be similar, but with rotating towards negative direction. This way every half plane going through w covering exactly k nodes is part of $h_-^w \cup h_+^w$, which altogether hit at most $2\rho_k$ edges. Since while turning a half plane around w, edges are getting hit or not hit one by one, and an edge at most 2 times, there can exist at most $2\rho_k$ hit edge sets with locally maximal cardinalities.

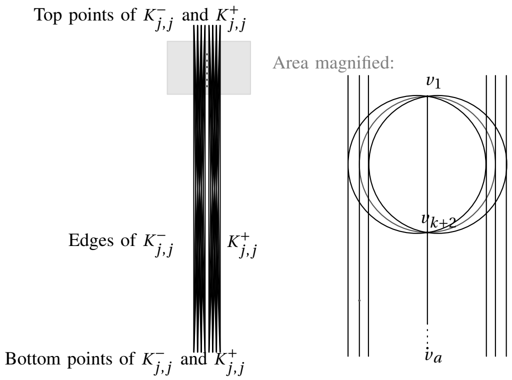

Top points of $K^-_{j,j}$ and $K^+_{j,j}$
Area magnified:
v_1
v_{k+2}
Edges of $K^-_{j,j}$
$K^+_{j,j}$
Bottom points of $K^-_{j,j}$ and $K^+_{j,j}$
v_a

Fig. 9: Sketch of a graph family where $|M_k| = \Theta\left(n^3\right)$ if $k = O(1)$.

Determining M_k^w can be done by simply turning the half plane around w. Checking currently hit edge sets, node sets and cardinalities can be done in a total $O(n\rho_0)$ time, list M_k^w can be created in $O(\rho_k^3)$. ■

Proof of Lemma 24: Trivially, every element of M_k^1 contains at most ρ_k elements. Based on Claims 34 and 35, lists M_k^w can be determined in $O(n^2\rho_0\gamma\log(n\rho_0)+n\rho_k^3)$ for all $w \in V$. M_k^1 can be calculated by merging the previous lists. $O(n^2\rho_k^2)$ comparisons have to be done, each has a complexity of $O(\rho_k)$. This means a total complexity of $O(n^2(\rho_0\gamma\log(n\rho_0)+\rho_k^3))$, completing the proof.

H. Connection Between Seesaws and Apples

It is known that the inverse of the stereographic projection from the North Pole, which maps geometric objects from the plane to the surface of the sphere has the property that the image of lines and line segments on the plane are great circles and geodesics on the sphere, respectively (corollary of Theorem in [32]). Known this, if we projected our network topology (back) to a spherical surface, and if we defined the spherical apple and spherical seesaw data structure, we could see that a spherical seesaw M_k^w is just a spherical apple $M_k^{w,z}$, z being the North Pole, thus sweeping through a spherical apple is the same thing as tilting a spherical seesaw, meaning that on the sphere there would not emerge the need to treat separately these two connected data structures.

Extreme cases

I. Maximum Number of Maximal Failures

In the followings, we show that there are some networks for which the running times are asymptotically not significantly better than our non-parametrized bounds. This motivates the introduction of parameter ρ_k, which captures the properties of real-life networks.

Theorem 4. $\max_{\mathscr{N}}|M_k| = \Theta\left(n^3\right)$ *if $k = O(1)$, where $\mathscr{N}$ is the set of all networks on n points.*

Proof: The proof will be immediate from Cor. 38. ■

It turns out that $|M_k|$ is $O((k+1)nm)$ (Claim 36), and in case of some artificial network families $|M_k|$ is $\Omega((n-k)n^2)$ (Cor. 38). This means that for $k = O(1)$ the maximum number of maximal failures is $\Theta(n^3)$. The details are the following.

Claim 36. *The number of disk failures can be at most $O((k+1)nm)$.*

Proof: Since $|E_k| = O((k+1)n)$ (Thm. 2), there are $O((k+1)n)$ apples. For each apple $A_k^{u,v}$, $|M_k^{u,v}| \le O(m)$ (proof of Claim 20). Since $|M_k^2| \le \sum |M_k^{u,v}|$, this means that $|M_k^2|$ is $O((k+1)nm)$.

On the other hand, the number of seesaws $|S_k| \le n$, and each $|M_k^w| \le O(m)$, which gives $|M_k^1| = O(nm)$. Since $|M_k| \le |M_k^2| + |M_k^1|$, the proof is complete. ■

Claim 37. *There is a graph family for which $|M_k| = \Omega\left(n^3\right)$ if $k = O(1)$.*

Sketch of proof: On Fig. 9 we can see a sketch of a graph $\mathscr{G} = (V, \mathscr{E})$ for which $|M_0| = \Theta(n^3)$. It has a so long-drawn shape it cannot be drawn precisely in a paper.

Let l be a vertical line. Both on the right and left side of l let us take a complete bipartite graph called $K_{b,b}^+$ and $K_{b,b}^-$, respectively (see Fig 9). We locate the points of $K_{b,b}^+$ and $K_{b,b}^-$ as follows. For both $K_{b,b}^+$ and $K_{b,b}^-$ one class of nodes is located at the top and the other at the bottom such that their vertices are equidistant on a horizontal line (i.e for nodes w_i in top class of $K_{b,b}^+$, distance $d(w_i, w_{i+1})$ is constant c for all $i \in \{1, \ldots, b-1\}$, same for the other classes). Let the y coordinates of the top and bottom classes differ in a much larger scale than the former constant c. Let $e_{i,j}$ be the edge incident to w_i and the j^{th} node in the bottom class. There is a horizontal line l_h such that common points $p_{i,j}$ of edges $e_{i,j}$ for a given i and $j \in \{1, \ldots b\}$ and line l_h are located nearly equidistant on l_h, let this distance be denoted by c_2. Clearly, l_h can be chosen such that the abscissa of $p_{i+1,1}$ is the abscissa of $p_{i,b}$ plus c_2 (for all $i \in \{1, \ldots, b-1\}$).

Let $a \ge k+2$. Let us consider $v_1, \ldots, v_a \in V$, different nodes lying equidistant on l, such that about half of them are above line l_h while the rest is below it. Let the distance between these nodes be c_v, where c_v is chosen such that the distance between the leftmost edge of $K_{b,b}^+$ and $K_{b,b}^-$ to be $(k+1)c_v + \varepsilon$, where $\varepsilon \simeq \frac{c}{2}$. Finally, we perturb the location of every node to avoid degeneracies.

Due to this setting, considering apples $A_k^{v_i, v_{i+k+1}}$, while sweeping from left to right, edges are getting hit and unhit in turn, thus $|M_k^{v_i, v_{i+k+1}}| = \Theta(b^2)$ for every $1 \le i \le a-(k+1)$.

Let $V_a := \{v_1, \ldots, v_a\}$, and $\mathscr{E}_a$ be the set of edges induced by V_a.

Since for all element $m_i \in M_k^{v_i, v_{i+k+1}}$, $m_i \cap \mathscr{E}_a = \{\{v_i, v_{i+1}\}, \ldots, \{v_{i+k}, v_{i+k+1}\}\}$, we can deduct that there are no duplicates in these lists, i.e. $|M_k| \ge \sum_{1 \le i \le a-(k+1} |M_k^{v_i, v_{i+k+1}}|$, since there are other apples which we have not considered.

This means $|M_k| \ge (a-(k+1))\Theta(b^2)$. The proof can be completed by choosing both a and b to be $\Theta(n)$, for example $a \simeq b \simeq \frac{n}{5}$. ■

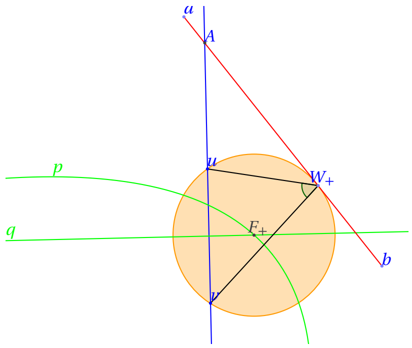


Fig. 10: Illustration for Subsec. J

Corollary 38. *The graph illustrated on Fig. 9. has $\Theta\left(n^3\right)$ maximal link k-node failures if k is $O(1)$.* ■

Although $|M_k|$ can be $\Omega(n^3)$, according to this paper in case of many real-life networks it is $O((k+1)n)$.

Computational Geometric Considerations

J. Determining $x_+(e)$ and $x_-(e)$ in $O\left(\gamma\right)$

Claim 39. *For any edge $e = \{a, b\}$ and node pair $\{u, v\} \subset V$, both $x_+(e)$ and $x_-(e)$ can be calculated in $O\left(\gamma\right)$.*

Proof: Let us concentrate on calculation of $x_+(e)$, because $x_-(e)$ can be determined similarly.

Extreme hitting disk of line: First, let us compute the leftmost hitting circle in $\mathscr{C}^{u,v}$ of a point W part of a given line ab. Let $\{A\} = uv \cap ab$ and let $x_+(W)$ be the abscissa of the center point of the leftmost hitting disk of W. Clearly, the function $x_+(W)$ is unimodal in both rays (R_+ and R_-) defined by line ab and point A (i.e., both R_+ and R_- are consisting of an interval where it is strictly monotone increasing and another interval where strictly monotone decreasing).

Let the two points on the line where the local minimum is reached be W_+ on the right side and W_- on the left side of uv. Let the centre point of disks $c(uvW_+)$ and $c(uvW_-)$ be X_+ and X_-, respectively.

X_+ is located on the Ox axis of the coordinate system of apple $A_k^{u,v}$. On the other hand, X_+ is located equidistant from u and line ab, thus it is on parabola p_u defined by point u and line ab. Similarly, let p_v be the parabola defined by v and line ab.

Since p_u and p_v can be characterized with quadratic expressions, which can be solved in $O(1)$, abscissa of X_+ can be found in constant time by determining their common root.

X_- can be determined similarly. Finally, the center point of the desired leftmost hitting disk is the one from X_+ and X_- with smaller abscissa.

Extreme hitting disk of line segment: When restricting the domain of a unimodal function, extremes can appear on the new boundary, thus for segment $[a, b]$, $x_+([a, b]) = \min(x(X_+), x(X_-), x(A), x(B))$.

Extreme hitting disk of the polygon of edge: For an arbitrary $e \in \mathscr{E}$ we consider two cases regarding the respective position of $[uv]$ and e^p.

In the first case segment $[uv]$ is entirely in the interior of e^p. Trivially, in this case $x_+(e) = -\infty$.

In the second case, segment $[uv]$ is not entirely in the interior of e^p. Since in this model a polygon of edge e^p consists of at most γ line segments, for determining $x_+(e)$ one can determine x_+ for all the line segments that take the minimum value of them in $O(\gamma)$ total time.

Finally, it remains to prove that one can distinguish between the former two cases in $O(\gamma)$. Recall that u and v have ordinates 1 and -1, respectively, both lying on the y axis. Let C be the set of ordinates of the intersection points of axis y and the line segments generating the polygonal chain. If $C \cap [0,1] \neq \emptyset$, than segment $[u,v]$ is neither entirely inside nor entirely outside of the polygon. Otherwise if $|C \cap (1,+\infty)|$ is an even or odd number, than $[uv]$ is outside or inside the polygon, respectively. Note that all the required operations can be done in $O(\gamma)$. ■