



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

Minimális költségű hálózati topológia bővítés regionális hibák ellen

TDK dolgozat

Készítette:

Hajdú Zsombor László

Konzulensek:

Dr. Tapolcai János

Dr. Pašić Alija

2019

Tartalomjegyzék

1. Bevezetés	4
2. Kapcsolódó munkák	6
3. A hálózat modellezése	7
3.1. Kitérő egy egyszerű példával	9
4. Számítógépes geometriai háttér	11
4.1. Hízlalás (Offseting)	11
4.2. Legrövidebb út keresés akadályok között	11
4.3. A sík particionálása zárt görbékkel	12
4.4. Implementációs lehetőségek	13
5. Javasolt algoritmus	13
5.1. Veszélyzónák meghatározása	13
5.2. Vágások kialakítása	15
5.3. Optimalizálás	16
6. Implementáció	18
6.1. Python	18
6.2. C++	20
6.3. A program futásának menete	21
7. Szimulációs eredmények	22
7.1. Gráfok paraméterei, és összefüggésük az eredményekkel	23
7.2. Konkrét gráfok, különböző r értékek	25
7.3. Konklúzió	28
8. Elképzelhető bővítések	29

Absztrakt

Modern életünk egyre jobban hagyatkozik az internetre, így a telekommunikációs hálózatok védelme egyre fontosabb kérdéssé válik. Az Internet kimaradások gyakran regionális szintű katasztrófák következményei, mint például egy földrengés, áradás, hurrikán vagy akár bombatámadás. A hálózat nagy kiterjedése miatt rengeteg olyan felhasználót is érint a hiba hatása, akik fizikailag messze vannak a katasztrófától. A dolgozatomban azt a kérdést járom körbe, hogy milyen módszerrel lehetne olyan hálózatokat tervezni, amelyek ellenállóak regionális hibáknak, vagyis miként garantálható, hogy egy katasztrófa ne tudja részekre szakítani a hálózatot? Milyen új összeköttetéseket lenne érdemes kiépíteni, hogy a hálózat “bombabiztos” legyen, és a közvetlenül nem érintett felhasználók ne maradjanak szolgáltatás nélkül?

A dolgozat első részében definiáljuk a probléma matematikai modelljét, amelyben a csomópontokat a síkon rögzített pontokon elhelyezett gráf csúcsainak, az optikai kábeleket pedig ezen csúcsok közt futó éleknek tekintjük. Ezek alapján kombinatorikus geometriai módszerekkel meghatározzuk a “veszélyzónákat”, amiket ha katasztrófa –földrengés, árvíz, vagy akár bombatámadás– érne, a hálózat több részre esne szét. Azután a feladatot részfeladatokra bontjuk, és geometriai algoritmusok segítségével keressük az így keletkező komponensek közötti új élek legolcsóbb útvonalait. Végül egy mohó heurisztikus algoritmus segítségével választjuk ki a legolcsóbb kerülőutak halmazát. Az így kiegészített új hálózat már regionális katasztrófáktól védett lesz, akárhol lépjenek fel azok.

A kezdeti eredmények alapján bizakodunk, hogy a heurisztika nem csak optimumhoz közeli, hanem optimális megoldást is nyújthat, illetve a modell további érdekes kiegészítésekre, és általánosabb következtetések levonására is alkalmas lehet.

Abstract

Our modern life is becoming more and more dependent on the Internet; thus the protection of telecommunication networks is arising as an even more critical issue. Internet outages are often caused by natural catastrophes, such as earthquakes, floods, hurricanes, or even by bombing attacks. Because of the size of these networks, a great deal of users are affected by the failure, even though they are physically far from the disaster area. In this study, I investigate the issue of creating regional failure resilient networks. In other words, how can one guarantee that a given disaster will not disconnect the network? Where should the operator lay down additional fiber, so the users that are not directly affected by the regional failure still are able to use the service, making the network “bomb-proof”?

In the first part of the study, we define the mathematical model of the problem, where nodes (or routers) are a graph’s vertexes positioned on fix points on the plane, and optical cables are edges running between these vertexes. Based on this definition and utilizing the tools of combinatorial geometry, we define the so-called “danger zones”. If such a zone is hit by a disaster such as an earthquake, flood, or bomb, the network could fall apart into multiple components. In the next step, we deconstruct the task into smaller problems and using already existing algorithms, we search for the geometric shortest paths of the routes of new edges connecting the components. Finally, we use a greedy heuristic to select the set of these shortest detour paths. After laying down new fibers along these paths, the network will be protected from regional failures wherever they should strike.

Our initial results seem promising, and the heuristic may provide not only a close-to-optimal but an optimal solution. We hope the model can be expanded and used to draw more general conclusions.

1. Bevezetés

Néhány, a közelmúltban megjelent cikk [28, 13, 15, 18, 9, 17, 23, 26, 36] rávilágított arra, hogy milyen védtelenek a távközlési hálózatok több, ugyanabban a fizikai régióban történő eszköz (mint például hálózati csomópontok vagy optikai kábelek) meghibásodása ellen. Ezeknek az egyszerre történő hibáknak több kiváltó oka is lehet, mint például földrengések, árvizek, forgószeklek, máskor egy tsunami vagy tornádó, de –ha tágabban értelmezzük a katasztrófa fogalmát– az esetleges emberek által okozott katasztrófa eseményeket, például bombatámadásokat is ide lehet sorolni. Ezen fajta meghibásodásokat –ahol egyszerre több csomópont vagy link esik ki egy összefüggő földrajzi területen– hívjuk regionális hibáknak.

Ezek a hibák kiváltképp veszélyesek, mivel a hálózat nem csak egy elemében tesznek kárt, hanem egyszerre több hálózati eszköz is kiesik, és ezzel jóval bonyolultabb helyzeteket állítanak elő, mint amikor csak egy optikai eszköz, vagy egy forgalomirányító eszköz hibásodik meg. Ezeknél a regionális hibáknál a hálózat szempontjából a katasztrófa sújtotta területeket ért kár után a második legnagyobb probléma az, hogy a hálózat sokszor részeire esik szét, így más felhasználók, akik egyébként elég távol lennének a katasztrófától is érzik a hatását a kieső szolgáltatások, vagy akár a teljes külvilág elérhetetlenné válik számukra.

Míg a hálózati hibák mindössze 20%-át okozzák előre eltervezett karbantartások és tesztelési műveletek, a maradék 80% két részre bomlik: 70% szimpla, különálló felszerelés meghibásodása, míg 30% az előbb említett, katasztrófák által okozott többszörös meghibásodás [21, 10]. Annak ellenére, hogy milyen nagy részét teszik ki a hálózati hibáknak, és hogy ezek ellen a hibák ellen intuitívan könnyebb tervezni (nem véletlenszerűen lépnek fel a hálózatban, hanem összefüggő területekre koncentrálnak), jóval kevesebb megoldást látunk a gyakorlatban a biztosításukra, mint az egyszerű meghibásodások ellen.

Példának találjuk a 2011-es japán földrengést, ahol a tenger alatt futó optikai kábelek megsérültek, így körülbelül 1500 telekommunikációs csomópontot ért meghibásodás, vagy a 2018-as attikai tüzelet Görögországban, ahol a környező területeken nem volt lehetőség a kommunikációra, és még a mentőcsapatok között is korlátokba ütközött minden kapcsolatteremtési kísérlet.

Az elhangzottak után joggal merül fel a kérdés tehát, hogy a hálózatot üzemeltető, illetve a hálózati infrastruktúrát biztosító cégek mit tudnak tenni annak érdekében, hogy ha megtörténik a baj, a hatását minimalizálni tudják

a közvetlenül katasztrófa sújtott területre. Milyen megelőző lépéseket lehet tenni, hogy a hálózat ellenállóbb legyen a regionális katasztrófákkal szemben?

A fentebb említett hibák elleni védekezésnek több lehetséges módja is van, például

- megosztott, ún. sötét kábel vásárlása,
- meglévő linkek vagy csomópontok további védelemmel való ellátása
- új optikai kábelek fektetése

Jelen dolgozatomban a legutóbbi módszert vizsgálom, azaz a kérdésfeltevés az, hogyan lehet egy hálózat topológiáját új élekkel úgy kiegészíteni, hogy az regionális hibáktól ellenállóvá váljon, a teljesség kedvéért azonban megemlítem az első két módszert is.

A meglévő kihasználatlan optikai kábelek vásárlása kedvező lehet abból a szempontból, hogy a vonalak már ki vannak építve, és a hálózat üzemeltetőjének csak meg kell vásárolnia és be kell üzemeltetnie őket. Azonban korlátokkal is jár, mint például a fix útvonalakkal és adatsebességekkel. Ezek a korlátozó tényezők nem mindig teszik lehetővé a hálózat csak ezzel a módszerrel való hibabiztossá tételét.

A második módszer a meglévő összeköttetések megerősítése. Ez bevett gyakorlat például az óceán mélyén fekvő kábeleknél, ahol az élővilág károsíthatja az összeköttetést [41]. Legnagyobb előnye, hogy a bővítés “házon belül” tud maradni, azonban még mindig az fixek útvonalak, és a hiba valószínűsége még így sem lehet teljesen 0-ra redukálni, azaz semmi nem garantálja azt, hogy egy katasztrófa a megerősítések után már nem fogja tudni tönkretenni a hálózatot.

Az új optikai kábelek kiépítése több szempontból előnyös: először is természetesen választhatjuk meg a kábelek útvonalait és sebességét. Ez lehetővé teszi, hogy minden kritikus régiót biztosan kikerüljünk, a megfelelő paramétereket kiválasztva. Látni fogjuk, hogy ha a hálózati berendezések egyéni (nem regionális katasztrófák által okozott) meghibásodásaitól eltekintünk, ezzel a módszerrel teljesen hibabiztossá válhat egy hálózat. Jogos kritika lehet azonban, hogy új összeköttetések kiépítése és üzemeltetése az előző két módszerhez képest jóval drágább lehet.

Visszatérve a dolgozat témájára, azt vizsgálom, hogy a problémát hogyan lehet optimális, vagy optimumhoz közelítő eredménnyel megoldani, hogy az új összeköttetések összköltsége a lehető legkevesebb legyen, amellet, hogy a

hálózat regionális hibáktól való védettsége garantált marad. Hogyan modellezhetünk hálózatokat és hibákat? Milyen módszerek, illetve algoritmusok állnak rendelkezésünkre, és ezeket hogyan tudjuk a gyakorlatba ültetni? Illetve ami a legfontosabb, milyen eredményekkel?

2. Kapcsolódó munkák

Bár ennek a dolgozatnak a regionális hibák elleni védelem a témája, meg kell említenünk az irodalmat, ami az egyszerű hibákkal –egyszeres él- vagy csomópont kiesésekkel–, vagy egy azonos hibacsoportba (lásd: SLRG-k, *Shared Link Risk Groups*) tartozó, nagy valószínűséggel együtt meghibásodó elemek kiesésével foglalkozik. Több oka lehet, hogy külön elemek egy SLRG-be tartoznak, például egy kábelen osztozik több összeköttetés, vagy földrajzilag is közel vannak egymáshoz. A terület egy extenzíven kutatott része a hálózattervezésnek [37, 42, 35, 40], rengeteg megoldással már évek óta találkozhatunk is különféle forgalomirányító protokollokban implementálva.

A regionális hibák elleni védettség –pontosabban a nem védettség, és a hálózatok sérülékenysége– a közelmúltban kapott egyre több figyelmet [31, 14]. Például egy cikk [16] a telekommunikációs hálózatok védtelenségére hívja fel a figyelmet, hiszen ezeknek a hálózatoknak a zavartalan működése már a mindennapjainkban elvárt, meghibásodásuk pedig mindenki számára komoly problémákat okozhat.

Több írás jelent meg arról is, hogy hogyan érdemes matematikailag modellezni a regionális katasztrófákat [26, 36, 3, 20]. Ezen munkák többsége azt feltételezi, hogy a regionális hibának valamilyen síkbeli alakja van, ami legtöbbször egyszerűen egy kör [28]¹.

A [7, 8, 2] cikkekben az ún. geodiverz forgalomirányítás kiterjesztésével foglalkoznak, ahol a kapcsolat kialakításakor használt utaktól követelik meg hogy kellő távolságra legyenek egymástól. Például [8] folyamokkal közelíti meg a problémát, míg [2] pontpárok közötti elérés biztosításával foglalkozik.

Az optimális megoldások témáját is taglalja pár cikk, mint például [20], ahol a linkek idő alatti terhelése is szerepet játszik. A probléma és hozzá hasonló problémák nehézségével [11] foglalkozik, míg [32, 30] a késleltetést, valamint a protokollokból adódó hibák kezelését helyezi előtérbe különböző hálózattervezési módszereket alkalmazva.

¹Ennek az angol irodalomban „circulal disk failure” modell a neve.

Pár munka külön vizsgálja az érintett elemek szerint a problémát. Míg In [27, 33, 9] például a szárazföldi összeköttetéseket vizsgál, találunk a víz alatti kábelekkel foglalkozó cikkeket is [5, 38].

[25] is kiemeli a regionális hibák elleni védelem fontosságát, és egy érdekes, tenger alatti kábelek fektetésével foglalkozó optimalizációs problémát vet fel, ahol egy beálló katasztrófa költségeit minimalizálják. A nem katasztrófa által okozott hibákat, azaz az egyszerű meghibásodásokat egyszerűen egy, a „társadalmi és üzemeltetői költségeket is figyelembe vevő” úton lefektetett plusz tartalék kábellel kezelik. Konkrét esetnek van felhozva a Földközi-tenger hibabiztossá tétele az általuk bemutatott lineáris programozást használó megoldással.

A topológiák k -összefüggőségéről említést tesz egy másik [34] cikk, ami kiemeli, hogy bár legtöbbször az összefüggőség van metrikaként használva, a hibák mégis sokszor egy területen lépnek fel, ezért valamilyen lokalitást kell figyelembe venni a vizsgálatokkor. Ezt hívják „régió alapú összefüggőségnek”, ahol bemutatják, hogy a RAÖ-val való számolás sok esetben jobb eredményeket produkál, mint az egyszerű régió alapú hibamodellek. Említést tesznek még különböző megkötésekről, illetve a probléma NP-nehézségéről.

[4] és [39] forgalomirányítás, valamint szoftver szempontból közelít a problémához, míg [1] egy sztohasztikus modellt ajánl fel a hálózat hibabiztosításának megoldására.

Ebben a dolgozatban azt az esetet vizsgáljuk, amikor a meglévő infrastruktúrán nem változtatunk, és a bővítés is csak és kizárólag új élekkel megengedett, és az egész hálózatra kiterjedő védelmet szeretnénk biztosítani.

3. A hálózat modellezése

Ahhoz, hogy a problémát formálisabb módon is kezelni tudjuk, a hálózatunkat egy $G = (V, E)$ irányítatlan gráfként képzeljük el, ahol minden $v_i \in V$, $|V| = n$, $i = 1 \dots n$ csomópontot egy csúcs, és minden összeköttetést egy $e_j \in E$, $|E| = m$, $j = 1 \dots m$ él reprezentál. Ezen felül a csomópontok el vannak látva a hozzájuk tartozó koordinátákkal, ami a síkbeli elhelyezkedésüket mutatja meg, azaz v_i -ket felírhatjuk a következőképpen: $v_i = \{i, v_i^p\}$, ahol i nem más mint az index (vagy a csomópont neve), v_i^p pedig a csomópont-hoz tartozó koordináta (egy $p_i \in \mathbb{R}^2$ pont). Ezeket a G gráfokat szokás még irányítatlan geometrikus gráfoknak is nevezni.

A modellünk kicsit egyszerűsít a valóságon azzal, hogy csak egyenes éleket engedünk meg csomópontok közt, azonban nagyobb kiterjedésű hálózatokra, mint például a gerinchálózatok, ez a feltevés jó közelítéssel a legtöbb esetben fent áll.

Az egyszerű, kör alakú hiba modelljét használjuk a katasztrófák felírására, ami túlbecsüli egy katasztrófa által sújtott terület nagyságát. A továbbiakban egyszerűen katasztrófaként fogok erre hivatkozni.

3.1. Definíció. Egy C kör által reprezentált katasztrófa akkor *talál el* egy e élet, ha annak két végpontja (e_a, e_b) között húzott szakasz elmetszi C -t. Az eltalált élek halmazát E_c -vel jelöljük.

Hasonlóképpen:

3.2. Definíció. Egy C kör által reprezentált katasztrófa akkor *talál el* egy v^p koordinátákkal rendelkező v csúcsot, ha $v^p \in C$. Az eltalált csúcsok halmazát V_c -vel jelöljük.

Ezek a katasztrófák hibákat okoznak, azonban mi csak egy bizonyos részükre vagyunk kíváncsiak, ezért definiáljuk, hogy mikor kritikus egy hiba a hálózatnak: ha a maradék csúcsok és élek nem összefüggő gráfot alkotnak.

3.3. Definíció. Egy C hiba akkor kritikus a hálózat szempontjából nézve, ha $V_c = (V \setminus V_c, E \setminus E_c)$ módon definiált gráf több komponensből áll.

Mi tehát azokra az esetekre fogunk koncentrálni, amikor a hiba úgy találja el a hálózatot, hogy az egymástól elszigetelt komponensekre bomlik.

Maga a feladat így adott: az összes elképzelhető C katasztrófától meg kell védenünk a hálózatot. Amint már előbb említettük, az új élek behúzására koncentrálnunk. Mivel a C által lefedett területeken fontosabb dolog is van a katasztrófa beálltakor, mint hogy elérhető legyen a hálózat, a megoldás során arra koncentrálnunk, hogy egymástól távoli, katasztrófa által nem sújtott csomópontok zavartalanul tudjanak kommunikálni egymással. Továbbá feltesszük, hogy egy új kábel kiépítése egyforma költségű mindenhol.

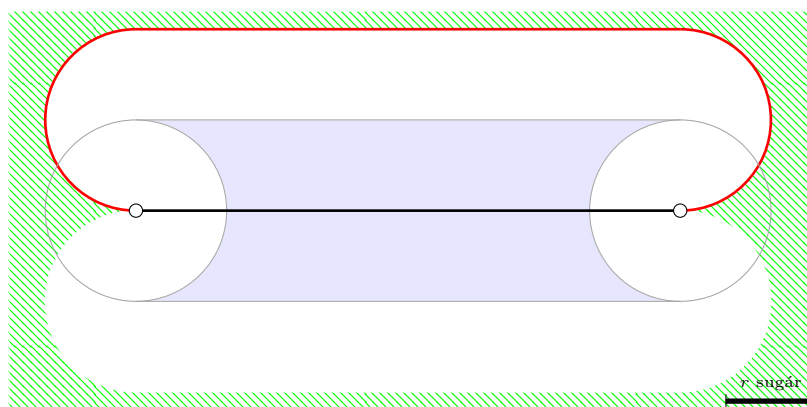
A célunk tehát az, hogy megtaláljuk a legkisebb költségű kibővítését a hálózati topológiának (ahol már megengedettek a tetszőleges görbék mentén futó élek is), ami már bármely, adott r sugarú C katasztrófát képes túlélni.

1. Probléma. Hálózati topológia hibabiztossá tétele: Adott r , a várt, tetszőleges C katasztrófa maximális sugara, és $G = (V, E)$ geometrikus irányítatlan gráf. Keressük azt az E_U élhalmazt, amelynek összköltsége minimális, és vele kiegészítve G -t a sík tetszőleges p pontjába eső katasztrófa ellen védett lesz a hálózat, azaz nem tud kritikus hibát okozni.

3.1. Kitérő egy egyszerű példával

Először vessünk egy pillantást a legegyszerűbb esetre, amikor a hálózatunk mindössze két csomópontból, és a közöttük futó élből áll. Ekkor a veszélyes pontokat könnyű meghatározni: ha a hiba középpontja bármely ponttól maximum r távolságra fekszik, akkor a maradék 1 pont definíció szerint összefüggő marad, így a pontok körül r távolságban biztos nem lesz kritikus hiba. Hasonló helyzet áll fent akkor, ha a csúcsoktól és az éltől is nagyobb mint r távolságra fekszik a hiba középpontja, hiszen ekkor érintetlenek maradnak az elemek.

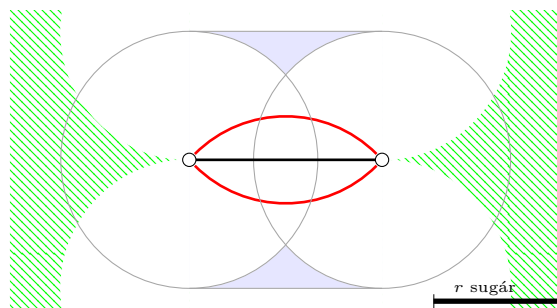
Akkor történik kritikus hiba, ha a hiba csak a linket érinti. Ekkor a zónát meghatározhatjuk úgy, hogy az él kör r sugárban egy „virslit” rajzolunk, és a végeiből kitöröljük a köröket. Az így kapott részt elnevezzük veszélyzónának, és ez az elnevezés a későbbiekben általánosabb értelemben is használni fogjuk.



1. ábra. Általános eset. A kék rész jelöli a veszélyzónát.

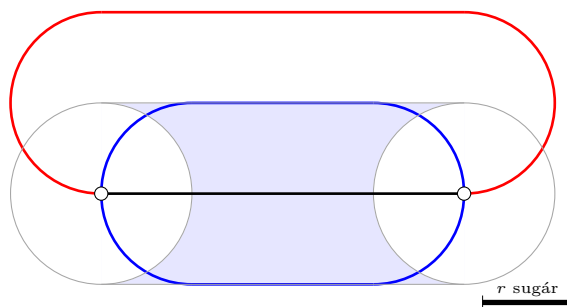
Könnyen láthatjuk, hogy a legrövidebb, veszélyzónát kikerülő út a -ból b -be párhuzamosan fog futni az eredeti éllel, azonban a végpontokból kiindulva

és beérkezve tesz egy-egy félkört. Az összköltség ekkor $|e| + r^2\pi$ lesz, ahol $|e|$ az él hossza. Ennél a hosszabb élnél egyértelműen látszik, hogy ez a jó megoldás, azonban van egy másik módszer is az utak meghatározására.



2. ábra. Rövid él

Ha megvizsgáljuk azt az egyszerű gráfot, ahol ugyan úgy két csomópont van, viszont közelebb helyezkednek el azok egymáshoz, más a helyzet: jobban járunk, ha külön-külön védekezünk a veszélyzónák ellen 2 külön éllel. Ekkor mindössze $4r\pi \frac{\arcsin(|e|/r)}{180}$ költséggel kell számolni, igaz, hogy ez a képlet csak akkor helyes, ha a két csomópontba rajzolt r sugarú körök metszik egymást. Egyébként a költség $2|e| + r^2\pi$ lesz, ahogy az alábbi ábrán is látszik:



3. ábra. $4r$ hosszú él

Behelyettesítve megkajuk, hogy a két költség pontosan akkor lesz egyelő, ha a két pont közti távolság pontosan $4r$, ahogy azt a fent ábra is mutatja.

4. Számítógépes geometriai háttér

A problémáink nagy részének megoldására már léteznek algoritmusok, és implementációkban is bővelkedünk. A fejezetben azt vizsgáljuk meg, hogy pontosan milyen részfeladataival kell foglalkozni a feladatnak, illetve ezekre a részfeladatokra milyen algoritmusok léteznek már.

4.1. Hízalás (Offseting)

2. Probléma. Adott Z régió a síkon, illetve $r \in \mathbb{R}$. Keressük Z^* régiót, amit a következőképpen határozunk meg:

$$Z^* = Z \cup \{ p \in \mathbb{R}^2, \text{ amelyek } Z\text{-től legfeljebb } r \text{ távolságra vannak.} \}$$

Ez a részfeladat azért fontos számunkra mivel, ha a veszélyes területeket védeni akarjuk, minimum r távolságra kell a kerülőutakat kiépíteni, hiszen ha ennél közelebb fut egy él egy általa védett zónához, akkor annak a zónának az élhez r -nél kisebb távolságra lévő pontjai nem lesznek védettek (hiszen akkor az új él is kiesne).

A feladattal egy sokat kutatott területre mutat, igen nagy motivációja volt a kutatásnak a fúró és marógépek útjának kiszámítása ipari gépekben a gyártási folyamatokat megelőzően, vagy azok alatt. A fúrófejek is rendelkeznek egy " r " paraméterrel, hiszen nem 0 átmérőjűek a bitek, és a munkadarabot megformázó gép útját úgy kell megtervezni, hogy csak a szükséges részek legyenek eltávolodva/kilyukasztva.

Adottak tehát a régióink, és ezeket a régiókat r -el kell „meghízalni”, hogy utána legrövidebb utakat tudjunk számolni. Sajnos általánosan a probléma nehezen megoldható feladat (értsd: NP nehéz) [22]. Esetünkben azonban, amikor a kérdéses régiókat csak egyenes szakaszok valamint körívek határolhatják, már egyszerűbb dolgunk van, hiszen egyenes szakaszoknak és köríveknek is egyenesek és körívek határolják a „meghízalt” területét [29]. Erre a verzióra már létezik hatékony, polinom idő alatt futó algoritmus.

4.2. Legrövidebb út keresés akadályok között

3. Probléma. Adott $Z_1 \dots Z_k$ régiók a síkon (esetünkben ezek már a felhízalt régiók), és a valamint b pontok, amiket nem tartalmaz egyik Z_i régió

sem. Keressük azt a görbét, amely összeköti a -t b -vel, minimális költségű, valamint nem metszi el egyik Z_i régiót sem.

A legtöbb munka [19, 24] poligonok által meghatározott kikerülendő területekkel számol, azonban nekünk körívek is határolhatják a területeket. Egy frissebb munka [6] azonban ún. „spilnegon”-okat feltételez, ahol a poligon e élei kicserélhetőek a két csúcs közt futó e' görbékkel, ahol e és e' egy konvex területet zár közre. Ezt a problémánkat -mint később látni fogjuk, implementációs okok miatt- azonban egyszerűen a körök n oldalú szabályos sokszögekkel való helyettesítésével oldottuk meg, azaz esetünkben Z_i -k poligonok.

Szerencsénkre a feladatunk lényegi részének, azaz a legrövidebb utak keresésének és kiszámításának léteznek rendkívül gyors és könnyen kezelhető algoritmikus megoldásai. Rengeteget kutatott terület, hiszen a legrövidebb útvonalak tervezésétől a robotmunkások raktáráruházakban való mozgásáig számos területen felhasználhatóak az algoritmusok.

Maga a probléma megoldható a polinom idő alatt a geometriai Dijkstra algoritmussal, aminek futási ideje a legrosszabb esetben $O(n \cdot \log(n))$, ahol n Z_i , $i = 1 \dots k$ poligonokban az összes csúcspont száma.

4.3. A sík partícionálása zárt görbékkel

4. Probléma. Adott C halmaz, ami síkbeli görbéket tartalmaz. Keressük $A(C)$ elrendezést, a sík olyan felosztását, ami 0, 1 és 2 dimenziós cellákat, más néven pontokat, éleket és lapokat tartalmaz. A C belüli görbék metszhetik egymást, ezzel kialakítva pontokat, és közrezárhatnak területeket, ezzel kialakítva lapokat. A lapokat pedig élek határolják.

A felosztások a számítógépes geometriában sok helyen jelen vannak, és rengeteg felhasználási módjuk van.

A görbék nem feltétlenül x -monotonak, ami azt jelenti, nem biztos, hogy függvényként leírhatóak, illetve saját magukat is metszhetik. Az elrendezés meghatározása a következő módon történik: A C belüli görbéket először szétbontjuk x -monoton görbékre, és ezeknek az új görbéknek a halmazát C' -nek nevezzük el. Vegyük észre, hogy x -monoton görbéknek nem lehet önmagukkal metszéspontja. A következő lépés a C' -beli görbék metszéspontjainak

meghatározása, és a metszéspontok mentén való szétbontása új görbék C'' halmazára. Ebből a halmazból már könnyen meghatározhatunk egy síkgráfot, aminek a csúcsai a metszéspontok, és az élei pedig a C'' -beli görbék.

Könnyen látható, hogy $A(C) = A(C'')$, így a kapott síkgráfból könnyen meghatározhatjuk a felosztást. A fő ötlet az, hogy minden élet két „fél-él”-re bontunk, hasonlóképpen amikor egy irányítatlan gráfból irányítottat képzünk. Minden fél-éltől balra fog elhelyezkedni a lap, amit határol, és a fél-éleken egy lap mentén körvejárva egyszerre a határoló csúcspontokat is megkapjuk, ezzel egy poligont meghatározva².

A mi esetünkben csak x -monoton görbék (egyenesek) lesznek a felosztást alkotó C halmaz belüli görbék, és az eszközt arra fogjuk használni, hogy könnyen meg tudjuk határozni a potenciálisan veszélyes pontok halmazait, pontosabban az ezeket a területeket reprezentáló poligonokat.

4.4. Implementációs lehetőségek

A fejezetben tagolt problémák megoldására rengeteg kész algoritmus, illetve keretrendszer áll rendelkezésre. Mi az első két problémára (A hízlalás és leg-rövidebb utak kérdése) találtunk egy egyszerű eszközt, a Marcello Kallmann által írt SIG-et, ami lehetővé teszi két pont két pont közötti, különálló régiókat kerülő útvonal kiszámítását, ráadásul a hízlalás is implementálva van a csomagban. Ezt használjuk tehát az útkeresésre.

A felosztás megvalósításához rendelkezésünkre áll a CGAL könyvtár „2D arrangements” csomaga, amivel hatékonyan lehet egyenesek és görbék mentén felosztani a síkot, valamint meghatározni a különálló területeket. A partícionálásra ezt az eszközt használtuk.

5. Javasolt algoritmus

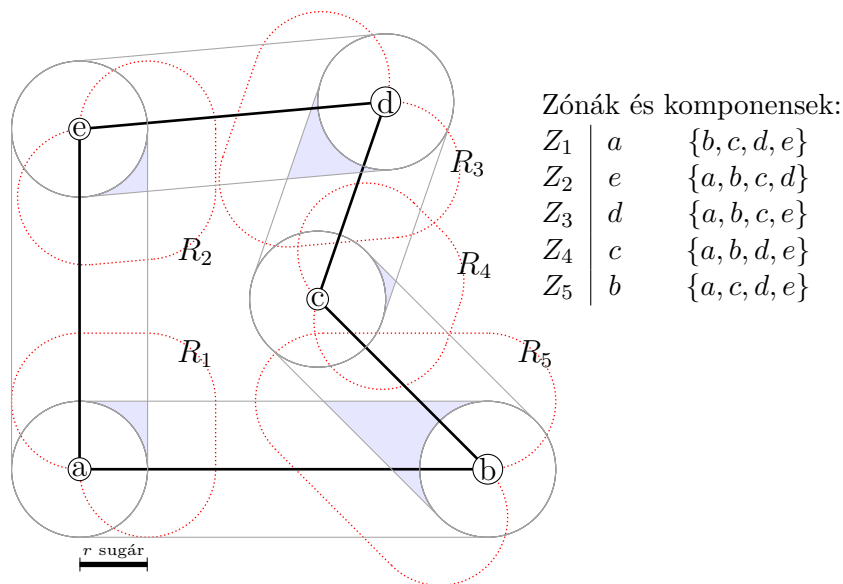
5.1. Veszélyzónák meghatározása

Mik tehát a “veszélyes pontok”, amiket kezelni kell a probléma megoldása során? Ezek a pontok összefüggő régiókat fognak alkotni: könnyen belátható,

²További információ a felosztásokról: https://doc.cgal.org/latest/Arrangement_on_surface_2/index.html

hogy egy csúcs akkor és csak akkor esik ki, hogy ha a katasztrófa középpontja tőle maximum r távolságra van. Ugyanezt elmondhatjuk az élekről is. Kérdés tehát, hogy hogyan lehet meghatározni a pontokat amiken, ha katasztrófa keletkezik, a hálózat meghiúsodik.

Ahhoz, hogy hatékonyan meg tudjuk határozni a veszélyes pontok által alkotott régiókat, először felbontjuk a síkot a következő módon: minden egyes csúcs köré egy r sugarú kört rajzolunk, majd minden éllel párhuzamosan két-két szakaszt rajzolunk r távolságra. Ezek a körök és szakaszok a síkból különböző síkidomokat metszenek ki. Minden egyes síkidomra egy egyedi csúcs-él konfiguráció fog kiesni a hálózatból, és ha a megmaradó hálózat nem összefüggő, ezek a síkidomok lesznek a veszélyes régiók.



4. ábra. Egy egyszerű példa hálózat, ahol a veszélyzónák az kékre vannak színezve, és a felhízalt régiók piros szaggatott vonallal vannak határolva.

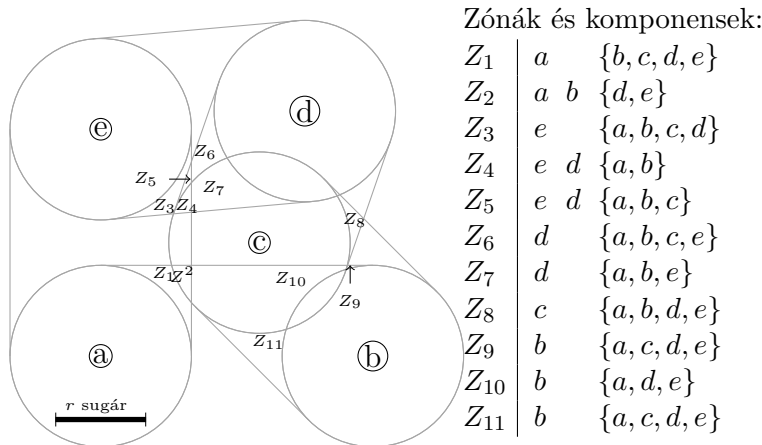
Ezeket a speciális síkidomokat, azaz veszélyes régiókat nevezzük el –a korábbiakra is hivatkozva– “veszélyzónának”, és $Z_1 \dots Z_k$ -val jelöljük őket.

Egy bizonyos r sugarú katasztrófa, ha bármely p pontjába sújt le egy Z_i veszélyzónának, a megmaradó hálózat ugyan úgy fog kinézni. A megmaradó 2 (vagy több) komponens között pedig úgy kell éle(ke)t húzni, hogy ezt a veszélyzónát minimum r távolságban elkerüljék. Ehhez kell először meghatározni a felosztásból a lapokat a síkon, majd a lapok közül meghatározni és

kiválogatni azokat, amelyekről elmondhatjuk, hogy egy belső pontba eső hiba kritikus lesz. Az ezt követő lépések során bizonyos lapok „hízalt” változatai³ között kell kiszámolni bizonyos csúcs-párok közötti legrövidebb utakat.

5.2. Vágások kialakítása

Fontos megemlíteni még azt is, hogy különböző r értékekre különböző konfigurációkat fogunk kapni a síkon. Például nagyobb r értékekere bonyolultabb lesz a felosztás, mivel több metszéspont keletkezik az alakzatok között. Ebből kifolyólag több lap lesz potenciálisan veszélyzóna, ahogy az az alábbi ábrán is látható:



5. ábra. Az előbbi példa egy nagyobb sugárral számolva. Figyeljük meg, hogy itt már megjelennek olyan veszélyzónák is (Z_4, Z_5), amik több komponensre bontják a hálózatot.

Mivel egy él maximum két komponenst tud összekötni, a veszélyzónákat tovább kell bontatunk úgynevezett kettős vágásokra a következő módon: tegyük fel, hogy egy Z_t veszélyzónát érő katasztrófa hálózatot több mint két komponensre robbantja szét. Ezek a komponensek legyenek rendre $H_1 \dots H_i$. A komponenseket az összes lehetséges módon két csoportba pakoljuk, ahol nincs üres csoport és az összes komponenst elhelyeztük.

Például egy három, (A, B, C) komponenst okozó veszélyzónát 3 kettős vágásra, azaz $A|BC$ -re, $B|AC$ -ra, és $C|AB$ -re bontjuk szét. Az eljárás t

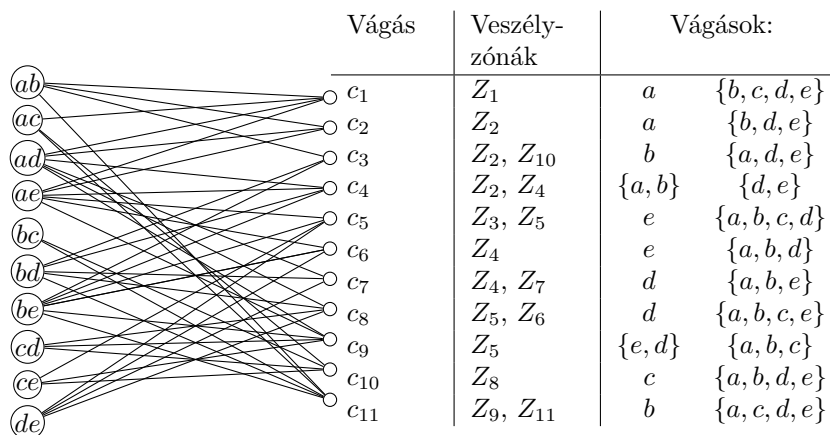
³Az ábrán a piros szaggatott vonallal jelölt területek

különálló komponens esetén ez összesen $2^{t-1} - 1$ db vágást fog eredményezni. Úgy is gondolhatunk erre a módszerre, hogy a különálló komponenseket egymással szembe, két oldalra válogatjuk azzal a céllal, hogy az oldalak közötti élek behúzásával javítjuk majd az eredeti veszélyzónát is. A vágásoknál eltároljuk, hogy milyen veszélyzónából keletkeztek.

Miután ezzel a lépéssel végeztünk, és rendelkezésünkre állnak $c_1 \dots c_m$ vágásaink. Azokat a vágásokat összevonhatjuk, amiknél ugyan az az eredményezett két csúcshalmaz-pár, és megfogalmazhatunk egy heurisztikát, aminek során a javító éleket válogatjuk be.

5.3. Optimalizálás

Először definiálunk egy (V_{pp}, V_v, E) páros gráfot, ahol a V_{pp} -ben található csomópontok a gráfba behúzható összes plusz élt reprezentálják (tehát pontpárokat, és a közöttük futó potenciális éleket tartalmaznak), $V_v = \{c_1 \dots c_m\}$ pedig a vágásokat jelölő csúcsok halmaza (amik a vágásokat, illetve a hozzájuk tartozó veszélyzónákat tárolják). Egy $x \in V_{pp}$ és egy $y \in V_v$ csúcs között futó élt akkor húzunk be, hogyha az a bizonyos $x = (a_x, b_x)$ él képes összekapcsolni a y csúcsban tárolt vágás két részét⁴.



6. ábra. Az optimalizáláshoz felépített páros gráf

⁴például ha egy vágás $\{abc|de\}$ csúcshalmazokból áll, javító él lehet számára ad, ae, be , stb. viszont ab, de , esetleg af már nem.

Fontos figyelni a terminológiára, ugyanis bár V_{pp} elemei az optimalizálás-hoz használt páros gráfban *csúcsok*, azonban az eredeti topológiában *csúcs-párok között húzható éleket* jelentenek. A heurisztika a páros gráfon fog műveleteket végzi el a következő, alapvetően mohó módon:

Heurisztika

1. **lépés** Számoljuk ki minden egyes $x \in V_{pp}$ csúcsra a költséget a következőképpen: minden olyan veszélyzónát vegyük be a kikerülendő zónák Z_H halmazába, amelyet $y \in N(x)$ csomópontok tárolnak, és (x, y) él érvényes, ahol $(N(x))$ a páros gráfban x szomszédjait jelöli. Számoljuk a legrövidebb utat az x által reprezentált a és b csúcs között, ami r távolságra kikerüli az összes $Z \in Z_H$ veszélyzónát, és legyen ennek az útnak a költsége x költsége is egyben.
2. **lépés** Válasszuk ki azt az $x \in V_{pp}$ csúcsot, amiből a legtöbb érvényes él indul ki a páros gráfban! Ha több ilyen létezik, azt az x -et válasszuk, aminek a legkisebb az előző lépésben meghatározott költsége. Jelölje ezt x_c .
3. **lépés** Tároljuk el x_c -ot a beválogatott H_x csúcsponatok halmazába, és érvénytelenítsük az összes olyan éleket, ami $N(x_c)$ -ből indul ki
4. **lépés** Ha van még érvényes él, folytassuk az 1. lépésnél, egyébként STOP.

A heurisztikus algoritmus első része után rendelkezésünkre áll H_x halmaz, ami a topológiába beválogatott éleket (azaz $x \in V_{pp}$ csomópontokat) tárolja. Ennek a halmaznak az összköltségén még javíthatunk, mivel $N(H_x)$ beli y vágások közül könnyen lehet, hogy több $x_i = (a_i, b_i) \in V_{pp}$ él is védi azokat. Ekkor a következő lépésekkel válogatjuk ki a legjobb lehetőségeket:

5. **lépés** Válasszunk egy olyan $y \in N(H_x)$ csomópontot, amire igaz, hogy $|N(y)| \geq 2$. Legyen ez y_c .
6. **lépés** y_c minden lehetséges x_i szomszédjára megnézzük, hogy hogy változna a költség a többi úton, ha csak x_i védené a kérdéses vágást. A legnagyobb javítást jelentő x_i csúcsot jelölje x_{max} .
7. **lépés** $\forall x \in N(y_c) \setminus \{x_{max}\}$ csúcsra újraszámoljuk a költséget úgy, hogy $N(x) \setminus \{y_c\}$ által reprezentált veszélyzónákat kerülje csak az útvonal.

8. lépés Ha $\exists y \in N(H_x)$ amire $|N(y)| \geq 2$, folytassuk az első lépésnél, egyébként STOP.

A végén maradt H_x halmaz csomópontjaiban el lesznek tárolva a költségek, és $\forall x \in H_x$ -re $N(x)$ fogja jelölni az általa javított vágások halmazát.

6. Implementáció

Az algoritmus megvalósításához két nyelvet, a Pythont és a C++-t választottam:

6.1. Python

Azért esett a Python nyelvre a választás, mivel a magasszintű nyelvi elemeknek, és a megannyi, munkát segítő könyvtárnak hála gyorsan lehet benne jól olvasható kódot írni. Nemrégiben igen felkapott és népszerű nyelv, Egyes kutatások szerint a fejlesztők kb. 20%-a gondolja a legjobb, legkényelmesebb nyelvnek.

A fájlműveletek nagy részét, illetve a adatmodellt – azaz a Veszélyzónákat reprezentáló **DangerZone** osztályt, a Vágásokat reprezentáló **DisasterCut** osztály, az ezeket tároló és kezelő **DangerZoneList**, **CutList** osztályokat –, és az optimalizáláshoz szükséges páros gráfot (**BipartiteDisasterGraph** osztály) mind Pythonban írtam meg. Az optimalizáló heurisztika is ezen a nyelven lett megvalósítva, szintén az olvashatóság kedvéért. Az eredmények láthatóvá tétele érdekében a naplózás (*elterjettebben logging*) is történik valamint az egyszerűbb feladatok megoldása is pythonban lett implementálva:

Gráfok beolvasása

Két fajta bemeneti kiterjesztéssel birkózik meg a program, LGF és JSON formátumból tud beolvasni gráfokat. LGF esetén egy vele ekvivalens JSON leíró is készíthető, mivel a C++-ban implementált másik részprogramnak csak az a megfelelő bemenet. A beolvasás a python *json* könyvtárával történik JSON esetén, LGF esetén pedig egy reguláris kifejezés dolgozza fel a fájlokat.

Gráfok tárolása, műveletek

Miután a beolvasás megtörtént, a program gerincét képző *networkx* könyvtár felelős a gráf kezeléséért. A *networkx* egy gráfok tárolását, kezelését biztosító, és rengeteg gráfokhoz kapcsolódó algoritmust megvalósító programkönyvtár, aminek a segítségével történik például a paraméterek kiolvasása, maximális foksámú csúcs meghatározása, és megannyi más feladat egy függvényhívással történő megoldása.

Veszélyzónák meghatározása

A sík felosztását követően rendelkezésünkre állnak a régiókat leíró poligonok, amiket a *shapely* könyvtár segítségével kezelünk. Roppant egyszerű a dolog: A poligonokból egy P belső pontot mintavételezünk, ez az ún. reprezentatív pont. Utána egyszerű vektorműveletekkel eldöntjük, hogy ettől a P ponttól mely élek illetve csúcsok találhatóak közelebb mint r . Ezeket a csúcsokat és éleket kitöröljük, majd a *networkx* segítségével eldöntjük, hogy összefüggő-e a maradék gráf.

Itt még fontos megjegyezni, hogy az olyan 2 fokú csúcsok esetében, ahol az élek közel 180 fokos szöget zárnak be és a katasztrófa beálltával izolált pont keletkezne, azt a parányi veszélyzónát ami így jön létre ignoráljuk. Ennek a motivációja egyrészt szimplán az, hogy a veszélyzóna közel r távolságra fekszik a ponttól, és a valóságban jóval nagyobb eséllyel érné ugyan úgy a pontot is katasztrófa, másrészt pedig, hogy ezek a veszélyzónák kiterjedése elhanyagolhatóan kicsi, így a valószínűség, hogy pont oda esik egy katasztrófa (mivel jelen modellünkben egyenletes eloszlást feltételeztünk) amúgy is 0 hoz közeli.

Ábrák generálása

A látható eredmények érdekében a *matplotlib* névre keresztelt, grafikonok és egyéb ábrák generálására alkalmas könyvtár függvényei köré írtam egy kis saját kódot, amivel az eredeti topológiát, veszélyzónákat, illetve a kerülőutakat lehet megjeleníteni.

6.2. C++

Hátulütője azonban, hogy –magasszintű volta, illetve az interpretáltsága miatt– lassan fut. Ezért a kevésbé számításigényes részeket kódoltam csak le benne, és a nehezebb feladatokat a C++ nyelven oldottam meg. Azért döntöttem mellette bizonyos feladatok megoldásánál, mivel a pythonhoz képest nagyságrendekkel gyorsabb futást eredményez, illetve hozzá hasonlóan ugyan úgy rendelkezik rengeteg nyílt forráskódú könyvtárral, amiket a fejlesztés során kicsit átalakítva, a feladat specifikus részeihez igazítva tudtam hasznosítani.

Sík felbontása

A sík felbontását az előbb említett CGAL könyvtár segítségével oldottam meg úgy, hogy a részfeladatot egy külön futtatható állomány végzi el, bemenetnek egy gráf topológiáját leíró JSON fájljal, illetve az r paraméterrel. Kimenet egy szimpla szöveges fájl, ami soronként tartalmazza a síkon a különálló régiókat. A JSON fájl alapján a köröket N oldalú ($N \geq 120$) sokszögekkel közelíti, és az éllel párhuzamos szakaszokat ezen sokszögek megfelelő csúcspontjaihoz illeszti, hogy a lebegőpontos aritmetikából adódó esetleges finom számítási hibákat ezzel megkerüljük. A gráf beolvasása a [nholmann/json](#) könyvtárral történik, majd a kiíratásig mindent – a görbék felvételét, metszéspontok számítását, majd a lapok meghatározását – a CGAL könyvtár végzi el.

Hízlalás, legrövidebb utak

A hízlalást és a legrövidebb utak számítását egy megosztott könyvtárba *shared library* fordítottam, amit a python által gyárilag biztosított *ctypes*⁵ könyvtárral tettem elérhetővé az eredeti kód számára. A könyvtár hozzáférést biztosít egy általam írt *PathPlanner* osztályhoz, ami a *SIG* által rendelkezésre bocsájtott eszközöket használja. Első lépésként a kikerülendő régiókat kell betölteni az osztályba, majd ezek után egy index halmazzal megjelölni a konkrét esetekben elkerülni próbált zónákat. Ugyanígy be kell állítani a kezdő és végpontot, majd egy hívás után visszakaphatjuk a költséget, illetve az utat. Szintén a lebegőpontos hibák miatt r értékét néhány

⁵<https://docs.python.org/3/library/ctypes.html>

esetben egy $\epsilon > 0$ számmal csökkenteni kell, mivel megeshet, hogy a hízlalt területeken belülré esik az egyik pont, így lehetetlen legrövidebb utat számolni. Azonban az ebből az ϵ -ból származó eltérés roppant kicsiny, ezért elhanyagoltuk a számításoknál.

6.3. A program futásának menete

Ebben a rövid fejezetben ismertetem nagy vonalakban a kód futásának menetét. A kód refaktorálás után elérhető lesz githubon, a [hajduzs/netext](https://github.com/hajduzs/netext) repo alatt.

1. **Betöltés.** Betöltjük a topológiát a memóriába, elkészítjük a networkx gráfot a további műveletekhez.
2. **Felosztás.** A topológia alapján elindítjuk a sík felosztását intéző alprogramot, majd amikor az végzett, beolvassuk a lapokat.
3. **Veszélyzóna számítás.** A lapok közül mindegyikről eldöntjük, hogy kritikus hibát okoz-e egy belső pontba eső hiba, azaz veszélyzóna-e az adott lap. A veszélyzónákat eltároljuk, az általuk eredményezett maradék gárffal.
4. **Vágások kialakítása.** A veszélyzónák alapján elkészítjük a kettes vágásokat
5. **A páros gráf létrehozása.** Konstruáljuk az optimalizáláshoz szükséges páros gráfot a vágások alapján, kiszámoljuk a kezdeti költségeket
6. **Heurisztika.** A fentebb már taglalt algoritmus alapján meghatározzuk az optimálisnak gondolt megoldást.
7. **Kimenet.** Az eredményeket külső fájllokba mentjük, majd az ábrák generálása után kilépünk.

Futásidő

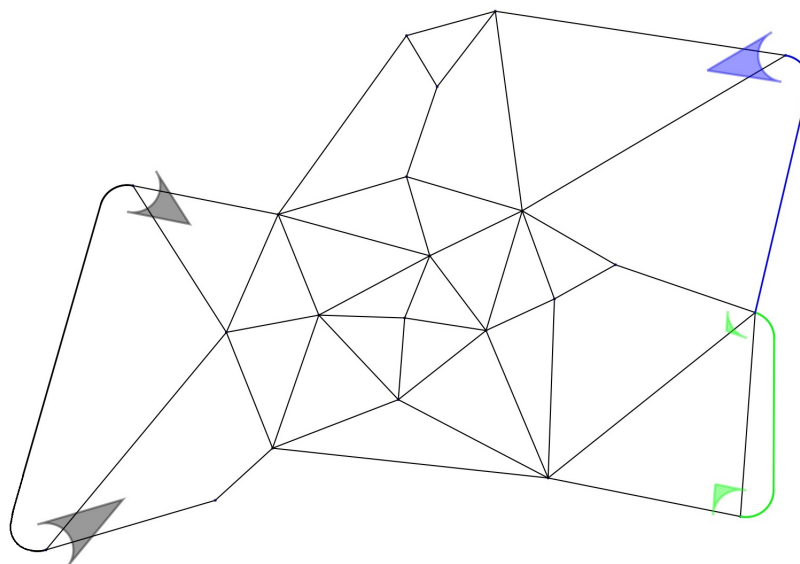
Érdekeség képpen még itt van egy táblázat arról, hogy bizonyos bemenetkerek milyen gyorsan futott az algoritmus.⁶

⁶Konfiguráció: CPU: i3-3110m 2.40Ghz, 8GB RAM, ubuntu linux

Gráf	Csomópontok száma	r	idő (s)
pelda	5	21.7	2.43
Funet-tz	23	24.3	10.37
optic-NA	38	60.6	32.30
gts-poland	25	25.35	64.53

7. ábra. Táblázat egy-két gráfra vizsgálva a futási időről. Láthatjuk, hogy nem feltétlenül a csomópontok száma határozza meg a futásidőt, inkább a topológia sajátosságai miatt „gondolkodik” sokat az algoritmus

7. Szimulációs eredmények



8. ábra. Az algoritmus által produkált élek az európai gerinchálózat kibővítésére $r = 104.5$ mellett. A képen a veszélyzónákat a nekik megfelelő színű élek védik.

Az elkészült algoritmus ellenőrzéséhez a konkrét gráfokat konzulenseimtől (ezeket [githubon](#) a graphs/ mappa alatt lehet megtekinteni), illetve [The Internet Topology Zoo](#) oldalról vettem kölcsön. Mielőtt konkrét eredményeket produkált volna a program, a következő paramétereket soroltam fel, kíváncsi voltam, hogy mit lehet mondani erről a heurisztikáról, illetve milyen változókat tudunk vizsgálni az eredmények megmagyarázásának érdekében.

7.1. Gráfok paramétere, és összefüggésük az eredményekkel

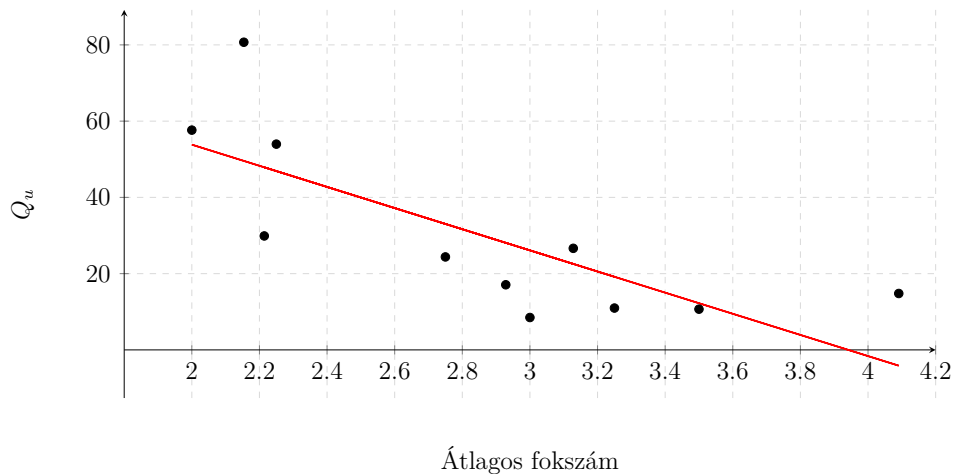
Jó célnak tűnt, hogy az algoritmus által talált élek össz. hosszát az eredeti élek össz. hosszához mérjük, és ezt próbáljuk különböző paraméterek függvényében magyarázni. A motiváció a célváltozó kiválasztása mögött az volt, a rendelkezésünkre álló gráfok külön-külön kiterjedésűek, ezért nem lehet csak az aggregált hosszúságot összehasonlítani akármivel is, hacsak nem ugyan azt a gráfot vizsgáljuk más-más r értékekre.

Általánosan elmondható, hogy a hálózat szélein jelenik meg több veszélyzóna, és hogy a perifériára szoruló csomópontok okozzák a legtöbb kritikus hibát.

A gráfok vizsgálata során az r értékét minden esetben a gráf pontjait befoglaló téglalap rövidebb oldalának 5%-ban határoztam meg. Az arányt az egyszerűség kedvéért Q_u -ként jelöljük a későbbiekben. A következő változók-ból próbáltam következtetéseket levonni Q_u -t illetően.

Átlagos fokszám

Az átlagos fokszámot azért tartottam érdemesnek megvizsgálni, ha külön pontokat nézünk, akkor 2 szomszédnál szinte biztos, három szomszédnál és ettől felfele nézve egyre kevésbé valószínű, hogy a pont izolálódik.



9. ábra. Q_u az átlagos fokszám függvényében. Megfigyelhetünk egy csökkenő tendenciát, azonban elég nagy a szórása az adatoknak.

Láthatjuk, hogy nagyjából igazolódni látszik a feltevés, miszerint az átlagos fokszám segít a dolgon. Azonban fontos megjegyezni, hogy nem annyira szoros a korreláció, látunk egy bizonyos szórást. Ez valószínűleg a gráfok különböző síkbeli topológiája miatt van.

Él/pontösszefüggőség

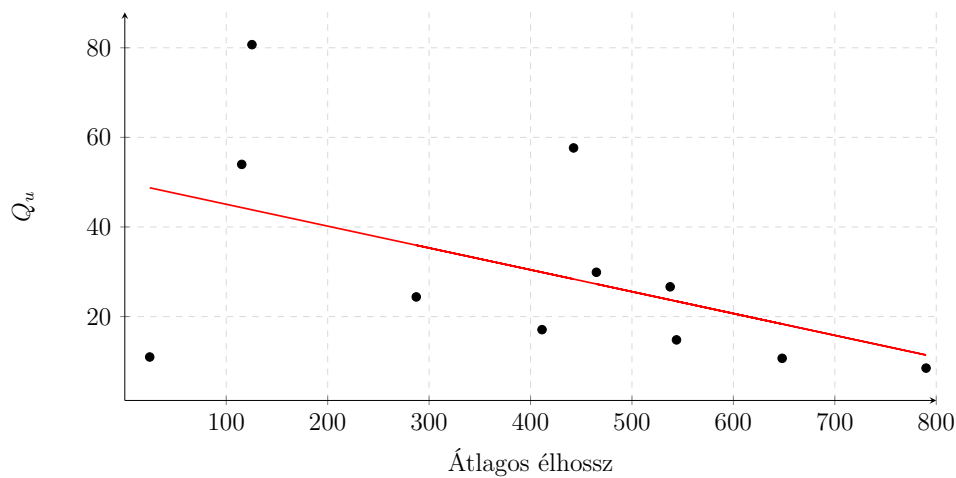
Sajnos a rendelkezésre álló gráfok mind 1-, illetve 2-összefüggőek voltak, az eredmények pedig nagy szórást mutattak, így az egyetlen leszűrhető tanulság az az, hogy nem látszik szoros kapcsolat az összefüggőség és Q_u között. *(Úgy is mondhatnánk, hogy nem függnek össze)*

A produkált eredményekből annyit láthatunk, hogy a 2-összefüggő topológiáknál átlagosan kisebb lesz az arány, azonban nem vonhatunk le túl sok következtetést több 1-összefüggő topológia híján, nem is beszélve $k \geq 3$ k -összefüggő gráfokra. Másrészt viszont az is igaz, hogy nagy k paraméterrel aligha létezik a valóságban topológia, szinte mindig találunk 1, de maximum 2 fokú csúcsokat a hálózatok szélein.

Érdekes észrevétel még, hogy ha az új élek darabszámát vizsgáljuk, akkor is látunk egy csökkenő tendenciát (igaz, ebben az esetben is nagy szórás mellett).

Élek átlagos hossza

Azért tartottam még érdekesnek megvizsgálni az élek átlagos hosszát, mivel ez a paraméter azt is jelezheti, hogy mekkora eséllyel esik ki egyszerre több él a hálózatból (hiszen minél hosszabbak az élek, annál inkább kisebb azoknak a területeknek az aránya az össz. területhez, ahol kritikus hiba keletkezhet)



10. ábra. Q_u az átlagos élhossz függvényében. Itt is láthatjuk, hogy csökken az arány, és nagyobb élekre kisebb szórást tapasztalunk.

Látható, hogy itt is létezik egy csökkenő tendencia, ami annak ellenére lép fel, hogy nagyobb távolságokat intuitívan nagyobb új élekkel kell áthidalni.

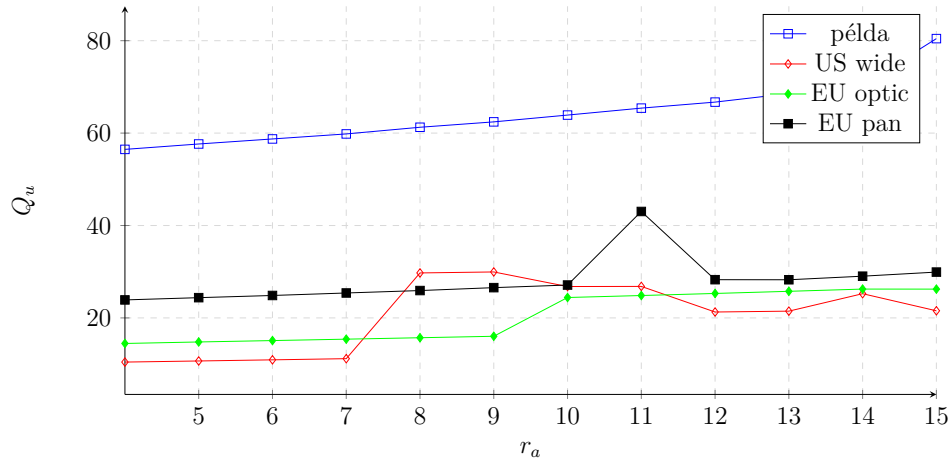
Az élek számát illetőleg az volt a tapasztalat, hogy r változásával nem nagyon változott egy adott gráfba behúzendó új élek száma (csak bizonyos gárfoknál nem volt elég a legkisebb próbált r -el kapott kezdeti érték, és a plusz élek (az összes vizsgált esetben mindössze 1 darab) is csak r nagy mértékben való növekedése után jelent meg.

7.2. Konkrét gráfok, különböző r értékek

Persze a legfontosabb paraméter, az r vizsgálatára is kitérünk. A következő részben arra vetünk egy pillantást, hogy konkrét gráfokra hogyan változnak az eredmények a katasztrófa sugarának változása függvényében?

A Q_u értéke az r függvényében

Intuitívan is következik, hogy nagyobb r értékekre több új kábel fog kelleni, és ez nagyon szépen látszik is az eredményeken. Fontos megjegyezni, hogy itt az r paramétert mér nem kvantitatívan, hanem arányosan vizsgáltuk: a topológiát bennfoglaló téglalap rövidebb oldalának r_a százaléka.



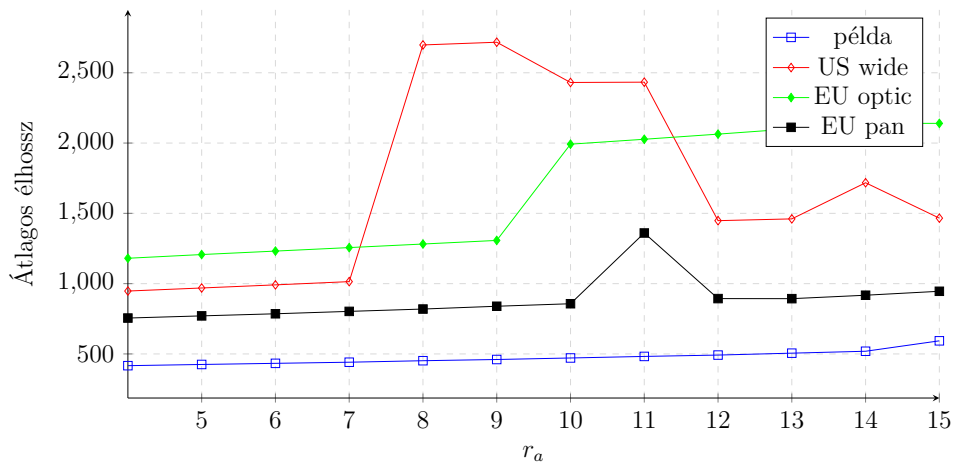
11. ábra. Q_u a r függvényében (arányosítva)

Nyilvánvalóan látszik a növekvő r értékekre a növekvő aggregált költség is, azonban megfigyelhetjük, hogy a növekedés –kisebb-nagyobb kiugrásoktól eltekintve, amik függenek a topológiától– lassú ütemű. Ez arra enged következtetni, hogy a bővítéseknek általában van egy fix költsége, ami a topológia elrendezéséből fakad, és ezután minél nagyobb katasztrófákat akarunk védeni, annál kisebb a relatív növekedés. Ezt azt jelenti, hogy a módszer r szerint jól skálázható.

Ez a „fix költség” akkor nyer egy kis világosságot, ha elképzeljük azt a szélsőséges esetet, amikor $r = 0$. Ekkor nem másról van szó, mint a geometrikus gráf 2-összefüggővé tételéről minimális költséggel. Ezen új éleknek az összköltsége egy mindenkor alsó korlátot is ad a probléma megoldására.

Az átlagos élhossz r függvényében

Bár azt gondolnánk, hogy minél nagyobb a védeni kívánt katasztrófa sugara, annál hosszabbak lesznek az élek is, látjuk, hogy nem minden esetben ez a helyzet:

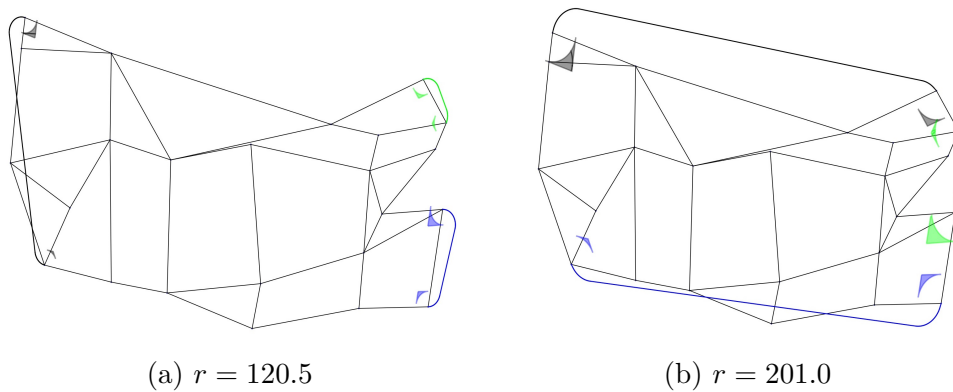


12. ábra. Az új élek átlagos hossza r függvényében (arányosítva)

Ez főleg a veszélyzónák elhelyezkedésétől függ. Jól látható például az amerikai gerinchálózaton, hogy mi okozza a kiugrásokat, és ezzel együtt ki-
 ütközik a heurisztika egyik hibája, hogy mindenképpen közös éleket fog be-
 válogatni bizonyos veszélyzónáknak, attól függetlenül, hogy azok milyen távol
 vannak egymástól. Ezt a következő szekcióban tárgyaljuk:

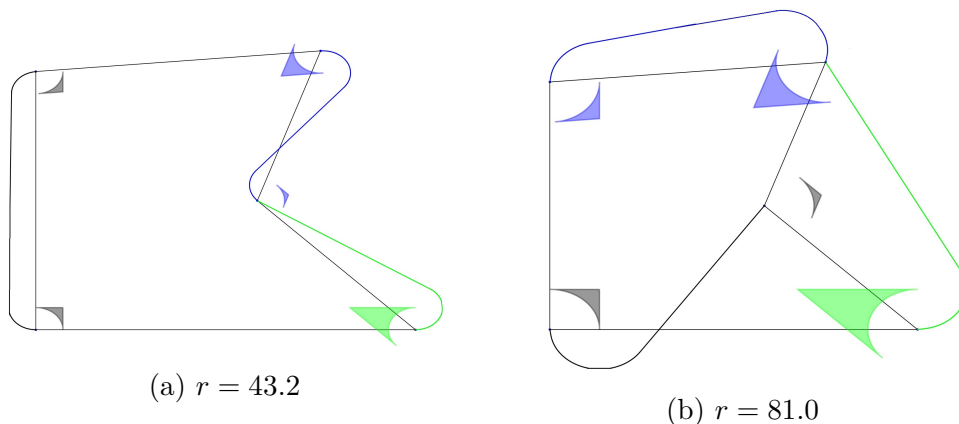
Az élek változása

Az előbb már említett amerikai hálózaton látszódik jól egy szembetűnő vál-
 tozás:



13. ábra. Az amerikai hálózat kiegészítése két eltérő r értékre.

Azonban nem minden topológiára jelent ez problémát. Az, hogy r növekedésével nagyban változhatnak az útvonalak, az 5. fejezetből már ismerős példa gráfon már jól látszik. Itt a különböző vágások kialakulása játszik szerepet a különböző élek beválogatásában.



14. ábra. Az példa hálózat kiegészítése két eltérő r értékre.

A heurisztika működési elvéből kifolyólag mindig a legtöbb még nem levédett vágást védő élet fogja behúzni. A lépések során kialakulhat az a helyzet, hogy a veszélyzónák nagy kiterjedésű területen vannak elszórva. Ekkor –mivel szigorúan a legtöbb szomszédal rendelkező csúcs lesz kiválasztva a páros gráfból– sajnos előállhat az a helyzet, hogy ahelyett, hogy a nyilvánvalóan optimálisabb megoldást, azaz a 2 vagy több külön éllel való védekezést választaná, ragaszkodik a közös élhez.

7.3. Konklúzió

A modell erős megkötéseket (nem csatlakozhatunk már meglévő élekhez) tesz, ami alapvetően megnöveli a hálózat kritikus hibáktól való védelmének költségét, és a heurisztika néhány nagy komplexitású esetben nem jó lépéseket tesz, ami nem feltétlenül vezet optimális eredményhez. Összességében azonban elmondhatjuk, hogy sikerült egy olyan módszer találni, amivel bármilyen r sugarú katasztrófától „védeni lehet” a hálózatot. A kisebb szépséghibák ellenére általában a hálózat eddig kiépített összeköttetéseihez képest a művelet nem igényel sokkal több élet sem hosszban, sem darabszámban mérve,

ráadásul a felhasznált algoritmusoknak hála nem érünk el túl nagy időbeli komplexitást sem.

8. Elképzelhető bővítések

Ebben a fejezetben röviden kitérek arra, hogy a kutatási projektben milyen egyéb lehetőségek vannak illetve, hogy a modellt/algorithmust milyen módon szeretném kibővíteni a közeljövőben.

Rövid, dupla élelkel való bővítés

Sajnos a program jelen állás szerint nem kezeli le azt az esetet, amikor jobban megéri 2 rövidebb élet húzni két csúcspont között, a dolgozat megírásának idejére nem jutottam el oda, hogy ez a funkció is működőképes legyen. Bár a valós hálózatok kiterjedése miatt egy esetleges katasztrófa által érintett terület „sugara” nem éri el általában a két egymáshoz legközelebb álló él távolságának negyedét, és amiatt, hogy általában egyszerre több zónát kell kikerülni, nem valószínű, hogy ez a speciális eset túl sokszor fent tud állni. Ezek ellenére azonban mégis szeretném implementálni azt is, hogy a modell tudjon számolni a rövid éleket javító dupla utakkal is.

Meglévő élekhez való csatlakozás

Az eredmények alapján látjuk, hogy néha olyan helyeken szelik át a hálózatot új élek, ahol már futnak régi összeköttetések. Erre megoldást kínál egy nemrég megjelent cikk [12], ami felhívta a figyelmet arra, hogy GPU shaderok segítségével rendkívül effektíven lehet legrövidebb utakat számolni, ráadásul lehetőség van a 2D síkon külön költségfüggvényt definiálni, ami azt jelenti, hogy síkrészekről függően megszabhatjuk az új utak költségeit bizonyos útvonalakon.

Ez azon kívül hogy valósághűbb modellezésre is lehetőséget nyújt (például egy síkságon vagy tengeren olcsóbban végig lehet vezetni egy optikai kábelt, mint hegyeken vagy lakott területen keresztül), azelőtt is kinyitja a kaput számunkra, hogy az új élekbe való becsatolást is, mint lehetőséget modellezni tudjunk. Ekkor mindössze annyi teendőnk lenne, hogy a már meglévő összeköttetések mentén lecsökkentsük a költséget zérusra.

Érdekesség: Az említett cikk szerzőitől származik az implementáció által használt SIG is, és egy-két másik cikk amit még felhasználtam a munka során. Ezúton is köszönöm nekik!

További algoritmusok, illetve heurisztikák vizsgálata

A fentebb bemutatott heurisztika bár első ránézésre jó eredményekkel szolgál, rengeteg módon lehet még erősíteni, új algoritmusokkal előállni és tesztelni őket. Érdeemes lenne más heurisztikus algoritmusokat is megvizsgálni, illetve megfontolandó lehetőség a bár exponenciális futásidővel rendelkező, de a probléma optimális megoldásával szolgáló algoritmus implementálása is, összehasonlítási alap gyanánt.

További gráfokkal való tesztelés

Bár érdekes dolgokat tudhattunk meg a heurisztika működéséről, mégis kicsit kevésnek éreztem a rendelkezésre álló topológiákból leszűrhető tanulságokat. A kód erőforrás-barátabbá alakítása után el tudnék képzelni még több vizsgálatot, akár véletlenszerűen generált gráfokra is. Ehhez érdemes lenne még több részt alacsony szinten (C++) implementálni, ahol párhuzamosítva is lehetne a számításokat végezni.

Hivatkozások

- [1] Anuj Agrawal, Vimal Bhatia, and Shashi Prakash. Network and risk modeling for disaster survivability analysis of backbone optical communication networks. *Journal of Lightwave Technology*, 37(10):2352–2362, 2019.
- [2] M. Waqar Ashraf, Sevia M. Idrus, Farabi Iqbal, and Rizwan Aslam Butt. On spatially disjoint lightpaths in optical networks. *Photonic Network Communications*, 36(1):11–25, Aug 2018.
- [3] Muhammad Ashraf, Sevia Idrus, Farabi Iqbal, Rizwan Butt, and Muhammad Faheem. Disaster-resilient optical network survivability: a comprehensive survey. In *Photonics*, volume 5, page 35. Multidisciplinary Digital Publishing Institute, 2018.
- [4] Yoshinari Awaji, Hideaki Furukawa, Sugang Xu, Masaki Shiraiwa, Naoya Wada, and Takehiro Tsuritani. Resilient optical network technologies for catastrophic disasters. *Journal of Optical Communications and Networking*, 9(6):A280–A289, 2017.
- [5] Cong Cao, Moshe Zukerman, Weiwei Wu, Jonathan H Manton, and Bill Moran. Survivable topology design of submarine networks. *Journal of Lightwave Technology*, 31(5):715–730, 2013.
- [6] Danny Z Chen and Haitao Wang. Computing shortest paths among curved obstacles in the plane. *ACM Transactions on Algorithms (TALG)*, 11(4):26, 2015.
- [7] Yufei Cheng, Deep Medhi, and James P. G. Sterbenz. Geodiverse routing with path delay and skew requirement under area-based challenges. *Networks*, 66(4):335–346, 2015.
- [8] A. de Sousa, D. Santos, and P. Monteiro. Determination of the minimum cost pair of D -geodiverse paths. In *The 2017 International Conference on Design of Reliable Communication Networks (DRCN 2017)*, Munich, March 8-10 2017.
- [9] Ferhat Dikbiyik, Massimo Tornatore, and Biswanath Mukherjee. Minimizing the risk from disaster failures in optical backbone networks. *Journal of Lightwave Technology*, 32(18):3175–3183, 2014.

- [10] Ferhat Dikbiyik, Massimo Tornatore, and Biswanath Mukherjee. Minimizing the risk from disaster failures in optical backbone networks. *J. Lightwave Technol.*, 32(18):3175–3183, Sep 2014.
- [11] B. Elshqeirat, S. Soh, S. Rai, and M. Lazarescu. Topology design with minimal cost subject to network reliability constraint. *IEEE Transactions on Reliability*, 64(1):118–131, March 2015.
- [12] R. Farias and M. Kallmann. Optimal path maps on the gpu. *IEEE Transactions on Visualization and Computer Graphics*, 2019.
- [13] Ori Gerstel, Masahiko Jinno, Andrew Lord, and SJ Ben Yoo. Elastic optical networking: A new dawn for the optical layer? *Communications Magazine, IEEE*, 50(2):s12–s20, 2012.
- [14] Teresa Gomes, János Tapolcai, Christian Esposito, David Hutchison, Fernando Kuipers, Jacek Rak, Amaro De Sousa, Athanasios Iossifides, Rui Travanca, Joao André, et al. A survey of strategies for communication networks to protect against large-scale natural disasters. In *8th International Workshop on Resilient Networks Design and Modeling (RNDM 2016)*, pages 11–22. IEEE, 2016.
- [15] M Farhan Habib, Massimo Tornatore, Marc De Leenheer, Ferhat Dikbiyik, and Biswanath Mukherjee. Design of disaster-resilient optical datacenter networks. *Journal of Lightwave Technology*, 30(16):2563–2573, 2012.
- [16] M Farhan Habib, Massimo Tornatore, Ferhat Dikbiyik, and Biswanath Mukherjee. Disaster survivability in optical communication networks. *Computer Communications*, 36(6):630–644, 2013.
- [17] Isil Burcu Barla Harter, Dominic Schupke, Marco Hoffmann, Georg Carle, et al. Network virtualization for disaster resilience of cloud services. *Communications Magazine, IEEE*, 52(12):88–95, 2014.
- [18] John Heidemann, Lin Quan, and Yuri Pradkin. *A preliminary analysis of network outages during hurricane Sandy*. University of Southern California, Information Sciences Institute, 2012.

- [19] John Hershberger and Subhash Suri. An optimal algorithm for euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.
- [20] Farabi Iqbal and Fernando Kuipers. On centrality-related disaster vulnerability of network regions. In *2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pages 1–6. IEEE, 2017.
- [21] Shrinivasa Kini, Srinivasan Ramasubramanian, Amund Kvalbein, and Audun Fosselie Hansen. Fast recovery from dual-link or single-node failures in ip networks using tunneling. *IEEE/ACM Trans. Netw.*, 18(6):1988–1999, December 2010.
- [22] In-Kwon Lee, Myung-Soo Kim, and Gershon Elber. Planar curve offset based on circle approximation. *Computer-Aided Design*, 28(8):617–630, 1996.
- [23] Xuelian Long, David Tipper, and Teresa Gomes. Measuring the survivability of networks to geographic correlated failures. *Optical Switching and Networking*, 14:117–133, 2014.
- [24] Joseph SB Mitchell et al. Geometric shortest paths and network optimization. *Handbook of computational geometry*, 334:633–702, 2000.
- [25] Dawson Ladislaus Msongaleli, Ferhat Dikbiyik, Moshe Zukerman, and Biswanath Mukherjee. Disaster-aware submarine fiber-optic cable deployment for mesh networks. *Journal of Lightwave Technology*, 34(18):4293–4303, 2016.
- [26] Biswanath Mukherjee, M Habib, and Ferhat Dikbiyik. Network adaptability from disaster disruptions and cascading failures. *Communications Magazine, IEEE*, 52(5):230–238, 2014.
- [27] Sebastian Neumayer, Alon Efrat, and Eytan Modiano. Geographic max-flow and min-cut under a circular disk failure model. *Computer Networks*, 77:117–127, 2015.
- [28] Sebastian Neumayer, Gil Zussman, Reuven Cohen, and Eytan Modiano. Assessing the vulnerability of the fiber infrastructure to disasters. *Networking, IEEE/ACM Transactions on*, 19(6):1610–1623, 2011.

- [29] H Persson. Nc machining of arbitrarily shaped pockets. *Computer-Aided Design*, 10(3):169–174, 1978.
- [30] Y. Prieto, J. E. Pezoa, N. Boettcher, and S. K. Sobarzo. Increasing network reliability to correlated failures through optimal multiculture design. In *CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies*, pages 1–6, Oct 2017.
- [31] Jacek Rak, David Hutchison, Eusebi Calle, Teresa Gomes, Matthias Gunkel, Paul Smith, Janos Tapolcai, Sofie Verbrugge, and Lena Wosinska. Recodis: Resilient communication services protecting end-user applications from disaster-based failures. In *18th International Conference on Transparent Optical Networks (ICTON 2016)*, pages 1–4. IEEE, 2016.
- [32] F. Robledo, P. Romero, and M. Saravia. On the interplay between topological network design and diameter constrained reliability. In *2016 12th International Conference on the Design of Reliable Communication Networks (DRCN)*, pages 106–108, March 2016.
- [33] Hiroshi Saito. Spatial design of physical network robust against earthquakes. *Journal of Lightwave Technology*, 33(2):443–458, 2015.
- [34] Arunabha Sen, Sudheendra Murthy, and Sujogya Banerjee. Region-based connectivity-a new paradigm for design of fault-tolerant networks. In *2009 International Conference on High Performance Switching and Routing*, pages 1–7. IEEE, 2009.
- [35] Arun Somani. *Survivability and traffic grooming in WDM optical networks*. Cambridge University Press, 2006.
- [36] Rodrigo Souza Couto, Stefano Secci, Miguel Mitre Campista, Kosmalski Costa, and Luis Maciel. Network design requirements for disaster resilience in iaas clouds. *Communications Magazine, IEEE*, 52(10):52–58, 2014.
- [37] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger. General availability model for multilayer transport networks. In *Proc. DRCN*, Lacco Ameno, Italy, October 16-19, 2005.

- [38] Weiwei Wu, Bill Moran, Jonathan H Manton, and Moshe Zukerman. Topology design of undersea cables considering survivability under major disasters. In *2009 International Conference on Advanced Information Networking and Applications Workshops*, pages 1154–1159. IEEE, 2009.
- [39] An Xie, Xiaoliang Wang, Wei Wang, and Sanglu Lu. Designing a disaster-resilient network with software defined networking. In *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, pages 135–140. IEEE, 2014.
- [40] Song Yang, Stojan Trajanovski, and Fernando Kuipers. Availability-based path selection and network vulnerability assessment. *Wiley Networks*, 66(4):306–319, 2015.
- [41] Jianan Zhang, Eytan Modiano, and David Hay. Enhancing network robustness via shielding. *IEEE/ACM Transactions on Networking (TON)*, 25(4):2209–2222, 2017.
- [42] Dongyun Zhou and Subramaniam Subramaniam. Survivability in optical networks. *IEEE Network*, 14(6):16–23, 2000.