



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Távközlési és Médiainformaticai Tanszék

# Szabály alapú útvonalépítési stratégiák nagy hálózatokban

DIPLOMATERV

*Készítette*  
Döbrei Gábor

*Konzulens*  
dr. Heszberger Zalán

2014. április 21.

## HALLGATÓI NYILATKOZAT

Alulírott *Döbrei Gábor*, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2014. december 18.

---

*Döbrei Gábor*  
hallgató

# Köszönetnyilvánítás

Nagyon sokat köszönhetek dr. Heszberger Zalánnak, aki alapdiplomás szakdolgozati, majd mesterdiplomás konzulensként nem csak elindított a kutatási irányba, de menet közben sok konzultációval, és baráti vezetéssel segítette fejlődésemet. A közös munka inspirációt is jelentett és megismertetett sok olyan eredménnyel, cikkel, amelyre egymagam nem találtam volna rá.

Sok közvetett segítséget és érdeklődést kaptam barátaimtól, évfolyamtársaimtól. Végül családom, elsősorban feleségem végtelen türelmére szükség volt a dolgozat létrejöttéhez.

# Tartalomjegyzék

Köszönetnyilvánítás	2
Kivonat	5
Bevezető	6
<b>1. Általánosított hálózatok és útvonalválasztási szabályok</b>	<b>9</b>
1.1. Történeti áttekintés . . . . .	9
1.2. Útvonalválasztás, routing algebrák . . . . .	11
1.3. Routing algebrák tulajdonságai . . . . .	14
1.3.1. Monotonitás, izotonitás és egyéb tulajdonságok . . . . .	14
1.4. Műveletek routing algebrák között . . . . .	15
1.4.1. A műveletek hatása a tulajdonságokra . . . . .	16
1.5. Példák algebráakra és kombinálásukra . . . . .	16
1.6. Összefoglaló . . . . .	17
<b>2. A hálózatkutatás legfontosabb modelljei</b>	<b>18</b>
2.1. Vírusterjedés komplex hálózatokban . . . . .	19
2.1.1. A vírusterjedés matematikai modellje . . . . .	19
2.1.2. Vírusterjedés, mint útvonalválasztási probléma . . . . .	20
2.2. Trendterjedés közösségi hálózatokban . . . . .	22
2.2.1. Trendterjedés, mint útvonalválasztási probléma . . . . .	22
2.3. Útvonalválasztás az Interneten . . . . .	24
2.4. Egyéb algebrák . . . . .	27
2.5. Összefoglaló . . . . .	28
<b>3. A modellezési keretrendszer</b>	<b>29</b>
3.1. Specifikáció . . . . .	29
3.1.1. A formátum . . . . .	30
3.1.2. Az optimalizálás függvénye . . . . .	31
3.1.3. A kimenet . . . . .	31
3.2. A szimulátor . . . . .	32
3.2.1. A matematikai struktúrák implementálása . . . . .	32
3.3. A szimulációs eredmények kiértékelése . . . . .	33

3.3.1. A vizsgálandó metrikák . . . . .	33
3.4. Valós hálózatok vizsgálata a keretrendszer segítségével . . . . .	37
3.5. Összefoglaló . . . . .	37
<b>4. Repülési hálózat vizsgálata</b>	<b>38</b>
4.1. Az adatok . . . . .	38
4.1.1. Repterek . . . . .	38
4.1.2. Repülési útvonalak . . . . .	39
4.1.3. Repülőtársaságok . . . . .	40
4.2. A szimuláció . . . . .	40
4.2.1. Az adatok előfeldolgozása . . . . .	40
4.2.2. A szimuláció menete . . . . .	41
4.3. Az eredmények . . . . .	42
4.3.1. A Shortest algebra . . . . .	42
4.3.2. A LeastHop algebra . . . . .	44
4.3.3. A ShortestLeastHop és a LeastHopShortest algebrák . . . . .	45
4.3.4. Az Összekötő-keresés- és Korai-elfogadó-keresés-LeastHop algebrák .	45
4.3.5. Az Összekötő-keresés-Shortest algebra . . . . .	45
4.3.6. A Korai-elfogadó-keresés-Shortest algebra . . . . .	46
4.3.7. Az algebrák értékelése . . . . .	48
4.4. Összefoglaló . . . . .	48
<b>5. Összefoglalás</b>	<b>50</b>
<b>Ábrák jegyzéke</b>	<b>52</b>
<b>Irodalomjegyzék</b>	<b>55</b>
<b>Függelék</b>	<b>56</b>
F.1. Számításelméleti alapok . . . . .	56
F.2. Az útvonalválasztás teljes modellje . . . . .	58
F.3. Tételek a routing algebrák témakörében . . . . .	58
F.4. A szimulátor technikai részletei . . . . .	59

# Kivonat

Az Internet rohamos fejlődésével az utóbbi időben mind nagyobb hangsúlyt kapnak a hagyományostól eltérő hálózatmenedzsment funkciókat megvalósító algoritmus kutatások. Ezek egyik legaktívabban művelt ága az útvonalválasztás kérdéseivel foglalkozik, és lényegében az erőforrások optimalizálást célzó klasszikus (főként legrövidebb utak megtalálására koncentráló) algoritmusok alternatíváinak kutatását tűzi ki célul. Az algoritmusok általános szabályok (policy) mentén alakítják ki a lehetséges kommunikációs útvonalakat.

A Diplomaterv célja olyan szabályalapú útvonalépítési stratégiák kutatása (szintetizálása / elemzése), mely mindenmellett, hogy alkalmas lehet valós kommunikációs hálózati alkalmazásra is, felhasználható más természetes vagy mesterséges / technológiai valós hálózatok működésének felderítésére.

A modern hálózatkutatás még nem adott megoldást arra a problémára, hogy egy hálózat útvonalválasztási szabályrendszerét hogyan határozhatjuk meg pusztán a kialakított utak megfigyelésével. Az útvonalválasztás matematikai modellezése [18] megfelelő alapot ad egy olyan szimulációs keretrendszer kifejlesztéséhez, amivel erre a problémára egy hiánypótló megoldást adhatunk.

Ebben a Diplomatervben leírok egy olyan szimulációs keretrendszeret, amely képes a nagyméretű valós hálózatok útvonalválasztási szabályrendszerét vizsgálni és alkalmasan választott új szabályokkal akár javítani is, valamint egy konkrét hálózatot meg is vizsgálok a keretrendszerrel: a repülőtársaságok útvonalválasztási szabályrendszerét modellezem.

# Bevezető

Az Internet rohamos fejlődésével egyre nagyobb igény mutatkozik a komplex számítógépes hálózatok megismerésére. Általános esetben a (kis) hálózatok struktúrája és működési mechanizmusai jól meghatározottak, hiszen a saját tervezésünk eredményeként jöttek létre és a vezérlés is a mi kezünkben van. Pontosan tudjuk, hogy egy hálózati csomópont melyik másik csomóponttal van kapcsolatban, ismerjük az összeköttetéseket és az útvonalválasztást meghatározó szabályokat is. Ha bármilyen módosítást szeretnénk eszközölni, vagy egy hibát szeretnénk kijavítani, azonnal (nagyon gyorsan) tudjuk, hogy hol kell beavatkozni.

Azonban egy olyan hálózat vizsgálata során, amelyet nem mi tervezünk, nagyon gyorsan szembesülik ilyen kérdésekkel, amiket csak nehezen és sok munka árán tudunk megválaszolni – mérésekkel, teszteléssel – és csak abban az esetben, ha a hálózat mérete még nem jelent problémát.

Az útvonalakat meghatározó algoritmusok általános szabályok (policy) mentén alakítják ki a lehetséges (kommunikációs) útvonalakat és bár a kapcsolódó irányadó paraméterek igen változatosak lehetnek, a problémakör egy meghatározó paramétere szerinti legrövidebb útvonalat szoktuk a legjobbnak tekinteni. Mégis, ha megvizsgáljuk az Internet magas szintű topológiáját<sup>1</sup> és a benne kialakult utakat, akkor azt vesszük észre, hogy ezek az utak nem az optimális megoldások, legalábbis nem a legrövidebbek. Mivel feltehetjük, hogy az anyagi haszon maximalizálása céljából az Autonóm Rendszerek<sup>2</sup> üzemeltetői racionális döntések révén építették ki pontosan ezeket az utakat, jogos kérdés, hogy pontosan milyen stratégia alapján tették ezt. Mi vezethet egy látszólag ésszerűtlen döntéshez? Általánosságban igaz az egyéb hálózatokra is, hogy az adott szituáció optimális útjai nem tűnnek racionálisnak.

Ha körbenézünk a világban, az informatikától távol eső területeken is rengeteg példát találunk olyan hálózatokra, amiknek nem értjük még a működését, nem tudjuk pontosan leírni a belső folyamatait. Számos példát találunk a biológiából, a szociológiából vagy a pénzügyi világból, ugyanakkor minden ilyen probléma vizsgálatát vissza lehet vezetni egy olyan modell vizsgálatára, ami általában képes kezelni magát a hálózat fogalmát és az azon történő útvonalak kialakulását / kialakítását. A vírusok által terjesztett betegség terjedése, egy ruhaviselet divattá válása, és a számítógépes hálózatok kommunikációs útjainak kialakulása modellezhető a gráfelmélet eszközrendszerével. Mindhárom esetben jól

<sup>1</sup>Nevezik még az Internet tartomány-szintű-, vagy AS-szintű topológiájának, gráfjának is.

<sup>2</sup>Autonóm Rendszer – Autonomous System (AS): önálló útválasztási tartomány, amelyen belül egyetlen, jól meghatározott útvonalválasztási szabály érvényesül.

meghatározható, elkülönülő csomópontok vannak, akik között terjed egy csomag, ami lehet információ vagy pl. a betegség maga. Egy modellen belül a csomópontok általában nem különböznek egymástól, és a köztük levő kapcsolatok különböző tulajdonságokkal rendelkeznek, amit az útvonalak kialakulása közben figyelembe is kell venni. Nem tételezünk fel különbséget két ember között, bárki meg tud betegedni. Magától értetődőnek látszik ugyanakkor, hogy levegőben terjedő vírusos fertőzés jóval lazább kapcsolaton keresztül is továbbterjed, míg egy vér útján terjedő betegség sokkal szorosabb kapcsolaton tud csak továbbterjedni: a kapcsolatokat súlyozni kell. Ha egy ember fogékonyabb egy betegségre, akkor a hozzá tartozó éleken nagyobb valószínűséggel terjed tovább majd a betegség. Ugyanilyen meggondolásból, két szomszédos router között a nagyobb sávszélességű úton továbbítjuk a csomagokat. Érdekes azonban, hogy egy divat elterjedését már nem tudjuk ilyen egyszerűen leírni, hiszen nem is igazán az a meghatározó, hogy mennyire befolyásos és ismert emberek reklámozzák, hanem az, hogy a társadalom felkészült-e már a befogadásra [8, 25].

A Diplomatervben a végső cél olyan policy primitíveket meghatározni, melyek összekapcsolásával jól lehet közelíteni a valós hálózatokat. Jelenleg sok megválaszolatlan kérdés van a BGP hálózat útvonalválasztásáról, melyek jó része onnan ered, hogy a hálózat csomópontjai, az AS-k nem fedik fel szabályrendszerüket. Ezen kérdésekre tudnánk válaszolni úgy, hogy nem a pontos szabályokat adjuk meg, hanem megmutatjuk, hogy adott policy primitívek felhasználásával majdnem ugyanolyan utakat kapunk, mint egy adott AS. Ez azért is lenne egy jó megoldás, mert lehet, hogy egy AS viszonylag bonyolult, sok erőforrást felemészítő szabályokat alkalmaz, pedig nagyon hasonló eredményeket érhetne el úgy is, ha az itt bemutatott p. primitíveket használná.

Az 1. fejezetben a számítógépes hálózatok policy-felderítésével foglalkozó szakirodalom áttekintés után egy általános hálózati- és routing modellt írok le, amely képes kezelni más tudományterületekről származó hasonló problémákat is. Az útvonalválasztás szabályrendszerét egy jól definiált matematikai struktúrával kell meghatározni, hogy definálni lehessen policy-k közti műveleteket, amelyekkel össze is tudunk kapcsolni policy-ket és tudjuk vizsgálni az kölcsönhatásukat.

A 2. fejezetben változatos problémákat és a hozzájuk tartozó routing policy-ket definiálok és leírom a hozzájuk tartozó matematikai struktúrákat, különös tekintettel olyan szabályrendszerekre, amelyeket a későbbi szimulációs vizsgálatoknál fel tudok használni.

A 3. fejezetben specifikálok egy olyan szimulációs keretrendszert, melynek segítségével a valós életből származó, tipikusan nagyméretű hálózatok útvonalválasztási szabályrendszereit képes jellemezni. Ehhez kidolgozok egy alkalmas pontozási rendszert, amely segítségével a szimulációs eredményeket értékelni tudom. Ennek részeként beazonosítom azokat a legfontosabb globális- és útvonalfüggő metrikákat, amelyekkel a lehető legpontosabban tudom mérni egy szimuláció pontosságát, valamint azt, hogy milyen mértékben tudnánk

javítani a megfigyelt hálózat útvonalválasztásán.

A 4. fejezetben a szimulációs keretrendszerrel elemzem, hogy egy valós hálózatban (repülőtársaságok útvonalkialakítása) milyen utakat határoznak meg az 1. fejezetből alkalmasan választott policy-k, és javaslatot teszek, hogy repülőtársaságok útvonalválasztását melyik policy-k keverékével lehet leg pontosabban modellezni, illetve javítani.

A Diplomatervben törekszem a szakmai közönség által érthető megfogalmazásra, általában a középiskolai matematikát meg nem haladó eszközrendszerrel dolgozom, az ennél komolyabb fogalmak, definíciók, tételek pedig megtalálhatók a függelékben. Mindazonáltal a számításelmélet, az algoritmuselmélet és egyéb magasabb szintű témakörök ismerete elengedhetetlen ennek a témanak a tárgyalása során.

A számításelmélet, az algoritmuselmélet és egyéb magasabb szintű témakörök ismerete elengedhetetlen ennek a diplomamunkának a megértése során, ugyanakkor minden fontosabb matematikai téTEL megtalálható a Függelékben.

## 1. fejezet

# Általánosított hálózatok és útvonalválasztási szabályok

Az Internet gerinchálózatát kialakító szabályrendszert nagy vonalaiban ismerjük. Az Internet AS-szintű topológiáját a BGP<sup>1</sup> határozza meg. Különböző eljárásokat ismerünk az Autonóm Rendszerek közti kapcsolatok feltárására a BGP-s routing táblák alapján, illetve az Internet router-szintű topológiájából, sőt az AS-ek routing-policy<sup>2</sup>-ját is tudjuk becsülni, ugyanakkor arra a kérdésre eddig még senki sem tudott választ adni, hogy miért éppen így alakultak ki a kommunikációs utak.

Ebben a fejezetben a szakirodalom rövid áttekintése után egy általános hálózati- és routing modellt mutatok be. Az útvonalválasztás szabályrendszerét egy jól definiált matematikai struktúrával – a routing algebrákkal – írom le, és definiálom a policy-k közti műveleteket, amelyekkel össze is tudunk kapcsolni policy-kat. Emellett egy alfejezet foglalkozik a policy-k különböző tulajdonságaival, melyek figyelembevétele fontos szempont a szimulációk megtervezésénél.

### 1.1. Történeti áttekintés

Az 1990-es évek végére a kutatók felismerték a tényt, hogy a BGP szintű Internet hálózatról nem tudunk szinte semmit. Rendelkezésükre állt néhány szolgáltató BGP-s routing táblája, de ez nagyon kevésnek bizonyult, hiszen a legtöbb AS közötti kapcsolat titkos gazdasági döntés révén született, így pontos képük nem lehetett a hálózatról. Ekkor kezdték el vizsgálni a különböző lehetőségeket, hogy hogyan lehetne feltárni ezt a rejtett hálózatot. Nem tűnt reménytelennek a helyzet, hiszen hálózat széle – a végpontok – nem tartoznak a szolgáltatókhöz, így egy tetszőleges számítógépről indított csomag útját végigkövetve értékes információkat nyerhettek. Természetesen a *traceroute*<sup>3</sup> futási eredményeit elemezve egy halom adatot kapunk, amit nehezen lehet csak feldolgozni, annál is inkább, mert az

---

<sup>1</sup>Border Gateway Protocol

<sup>2</sup>Útvonalválasztási szabályrendszer

<sup>3</sup>Egy számítógép-hálózati diagnosztikai eszköz, amivel az IP (Internet Protokoll) hálózaton haladó csomagok útját lehet követni.

egyes a mérések ismétlése során más és más útvonalon haladt át a követendő csomag. Képzültek szimulátorok, amik ügyesen kezelték a nehézségeket, és különböző heurisztikákat használtak, hogy csökkentsék a szükséges mérések számát [11], ugyanakkor még mindig nagy szakadék volt az eredmények és az elvárások között. Nem volt egyértelmű, hogy egy csomag miért épp az adott AS-eken keresztül ért célba.

Az ezredforduló után már sok új kutatás foglalkozott az Internet AS-szintű topológiájának feltérképezésével úgy, hogy az ehhez szükséges információt a kapcsolatokról még minden csakis a BGP routing táblákból szereztek. Volt azonban egy újfajta megközelítést is, ami az önálló AS-kapcsolatokat tárta fel az Internet router-szintű topológiájából [4]. Ez több szempontból is előnyös volt, mint a BGP-s routing táblákból kiolvasott topológia, hiszen

1. Nagyobb felbontásban látjuk az AS-szintű térképet (pl. látszódnak a többszörös kapcsolatok AS-ek között);
2. Látszódnak a BGP protokoll által aggregált – így az AS-szintű hálózatban eltakart – útvonalak;
3. Így már lehetőség volt azonosítani a határ-routereket<sup>4</sup>, aminek segítségével pontosabban karakterizálhatták az AS-en belül kapcsolatokat.

Az addigi eredményeket felhasználva már volt egy viszonylag pontos kép az Internet tartomány-szintű topológiájáról, amit tovább pontosítottak úgy, hogy az egyes AS-eken belüli szabályokat is feltárták, hiszen keveset tudtak még arról, hogy milyen routing policy-t használnak az AS-k. A BGP protokoll ugyanis lehetővé teszi az AS-eknek, hogy megválasszák az útvonalválasztási policy-jukat, ami alapján történik a csomagtovábbítás és az elérhetőségi adatok terjesztése az AS-en belül. Megmutatták, hogy az AS-k a többi szolgáltatók csak egy csoportjának hirdetik magukat, ami mögött valószínűleg traffic engineering<sup>5</sup> lehet. Pl. több Tier-1-es<sup>6</sup> AS is az ügyfeleit (közvetve vagy direkt) a peer kapcsolatain keresztül éri el a közvetlen customerei helyett. Ezenkívül a válogatott hirdetés szerint jóval kevesebb elérhető útvonal van az Interneten, mint azt az AS kapcsolati gráf mutatja (Ezért is volt jóval pontatlanabb a BGP routing táblák alapján elképzelt kép.) [23].

Miután már volt valamilyen fogalmunk arról, hogy az egyes AS-k hogyan működtetik hálózatukat, az AS-ek közötti kapcsolatok mélyebb megismerése következett. Adva van az Internetnek már megkülönböztetett router-szintű és AS-szintű topológiái, illetve felületesen már ismerjük az egyes policy-kat. Azt kezdték el vizsgálni, hogy a router-szintű topológiában mi lenne a legrövidebb út és ehhez képest mi az adott (valós) policy-út [21]. Több szempontból is vizsgálták a kérdést:

<sup>4</sup>Azon routereket, amelyek az AS szélén vannak és a többi AS-hez jelentik a kapcsolatot, határ-routernek nevezzük.

<sup>5</sup>Forgalomszabályozás: tudatos tervezéssel próbálják meg elkerülni azokat az eseteket, amikor szolgáltatás-kimaradás lép fel a túlterhelés miatt.

<sup>6</sup>A BGP hierarchiában a legmagasabb szintű szolgáltatók, ezeket követi a Tier-2 és a Tier-3.

- Mennyivel fújja fel a policy a shortest-utat<sup>7</sup>?
- Létezik-e  $S$  és  $D$  forrás-cél pár között egy  $I$  köztes pont, amire a  $d_{policy}(S \rightarrow I) + d_{policy}(I \rightarrow D) < d_{policy}(S \rightarrow D)$ ? (Ebben az esetben lehetne egy ilyen közbeiktatott ponttal javítani)
- A policy-k vajon a nagyobb AS-k felé terelik a forgalmat?

Ugyan abban az időben még csak a legrövidebb utakat vizsgálták és azokat is a multicasting<sup>8</sup> szempontjából, de azóta a kutatók felismerték, hogy olyan kérdések várnak még válaszra, amelyek alapvetően befolyásolják az egész Internet gerinchálózatát. Ezenkívül jelentősen megkönnyítené, illetve pontosabbá tehetné a téma körbeli kutatásokat, ha rendelkezésünkre állna egy olyan hálózati modell, amely nem csak a skálafüggetlenséget<sup>9</sup> tudja szimulálni, hanem a – gazdasági érdekek miatt titkolt – kapcsolatokat és útvonalválasztást is valóságban tudja modellezni.

Közöl húsz év óta foglalkoztatja a kutatókat a hálózatok kialakulása és a kialakult útvonalak, ugyanakkor az Internet AS-szintű topológiának vizsgálatakor arról nem állítanak semmit, hogy az egyes policy-kat miért használják. Sőt, ugyanígy nem tudunk szinte semmit a már említett más területekről származó problémák esetében sem.

Jelen Diplomaterv keretében egy olyan megoldást mutatok be, amely képes megbecsülni, hogy miért – milyen szabályszerűségek figyelembevételével – hoztak meg egyes döntéseket. Azzal a feltételezéssel éllek, hogy minden policy-t fel lehet építeni jól meghatározott, egyszerű primitívekből, illetve ezen primitívek azonosítása és változatos összekapcsolása révén meg fogjuk tudni mondani, hogy egy már kialakult hálózatot milyen komplex policy határoz meg.

## 1.2. Útvonalválasztás, routing algebrák

Az általános hálózati modellben egy véges, egyszerű és összefüggő  $G(V; E)$  gráfot használunk a hálózat megadására, ahol  $|V| = n$  és  $|E| = m$ . Az élek irányítottsága és súlyozása az adott problémától függ, minden kombináció megengedett. Legyen  $\deg(v)$  a  $v \in V$  csúcs fokszáma és legyen  $d = \max_{v \in V} \deg(v)$ . Egy  $s - t$  séta a csúcsoknak egy  $p = (s) = v_1, v_2, \dots, v_k (= t)$  sorozata, ahol  $k$  a séta hossza és  $(v_i, v_{i+1}) \in E : \forall i = 1, \dots, k-1$ . Ezenkívül egy kör olyan séta, ahol  $s = t$ .

Ahhoz, hogy meg tudjuk mondani, hogy két útból melyiket érdemes választani, definiálni kell egy preferencia sorrendet az utak között. Ehhez pedig az utak hosszára lesz

---

<sup>7</sup>Shortest-út: Azt a policy-t, amely a legrövidebb utakat választja, shortest-path (legrövidebb-út) policy-nak nevezzük.

<sup>8</sup>A multicast egy információtovábbítási mód, amikor egy üzenetet több célhoz szeretnénk eljuttatni egyidejűleg egyetlen forrástól.

<sup>9</sup>Egy hálózat skálafüggetlen, ha benne a fokszámeloszlás hatványfüggvényt követ. Az Internet hálózata tipikusan skálafüggetlen hálózat. (A kifejezés Barabási Albert László magyar származású amerikai matematikus nevéhez fűződik.)

sükség, amit egy metrikával, azaz távolságfüggvényel tudunk mérni. A metrikus tér fogalma egy (halmaz, függvény) párt jelent, ahol a függvény bármely két, halmazbeli elemhez egy nemnegatív valós számot rendel (vagyis a távolságukat méri).

**1.2.1. Definíció (Metrikus tér).** *A metrikus tér egy olyan  $(X, \delta)$  pár, ahol  $X$  egy tetszőleges halmaz,*

$\delta: X^2 \rightarrow \mathbb{R}_0^+$  pedig egy olyan nemnegatív valós függvény, melyre tetszőleges  $x, y, z \in X$  esetén:

1.  $\delta(x, y) = 0 \Leftrightarrow x = y$  (megegyezőségi tulajdonság)
2.  $\delta(x, y) = \delta(y, x)$  (szimmetria)
3.  $\delta(x, z) \leq \delta(x, y) + \delta(y, z)$  (háromszög-egyenlőtlenség).

Annak érdekében, hogy minél általánosabb lehessen a modell, minden  $e \in E$  élet egy tulajdonság-vektor jellemz, amely vektor minden dimenziójának értékeit különböző metrikák szerint adhatunk meg. Így definiálhatjuk az él hosszát, (sáv)szélességét, késleltetését, megbízhatóságát, sőt bármilyen nem tipikus tulajdonságot is (pl. szín, vagy egy időtől függő  $f(t)$  függvényt, stb.).

A metrikák fontos szerepet játszanak az útvonalválasztásban, hiszen ez alapján tudjuk a hálózati utak közötti preferenciát megadni, másnépp megfogalmazva megadja, hogy egy bizonyos tulajdonság alapján mekkora a költsége egy útnak. Az útvonalválasztás során a metrika lehet statikus, amikor egy előre rögzített elvet követünk végig, vagy lehet dinamikus, amikor a hálózat adott állapotától függően automatikusan változik. Azt, hogy milyen metrika szerint végezzük a routing-ot nevezzük *routing policy*-nak:

**1.2.2. Definíció (Routing policy).** *A routing policy egy olyan  $p_{st}^* = Pol(\mathcal{P}_{st})$  függvény, aminek az értelmezési tartománya a lehetséges  $s-t$  utak:  $\mathcal{P}_{st}$  és az adott policy-nak megfelelő legkedvezőbb utat adja vissza.*

Ahhoz, hogy ezentúl matematikailag is kényelmesen tudjuk kezelni a policy-kat, az ún. routing algebrák fogalmát kell használnunk, melyek az általános útvonalválasztó policy-k matematikai leírása [18, 19].

**1.2.3. Definíció (Routing algebra).** *Az  $\mathcal{A}$  routing algebra<sup>10</sup> egy teljesen rendezett félcsoport egy „végétlen elemmel”:  $\mathcal{A} = (W, \phi, \oplus, \preceq)$ , ahol  $W$  az élek súlyainak halmaza,  $\phi$  ( $\phi \notin W$ ) egy speciális végtelen súly, abban az értelemben, hogy azon az élen vagy úton nem lehet átmenni, a  $\oplus$  a súlyok egy kétváltozós kompozíció operátora, a  $\preceq$  pedig a súlyok összehasonlító operátora.*

Még pontosabban, a következő tulajdonságokat követeljük meg:

- $(W, \oplus)$  egy Abel-csoport:

---

<sup>10</sup>Az algebra elnevezés arra utal, hogy – mint később látni fogjuk – műveleteket lehet végezni ezen objektumokon.

- *Zárt:*  $w_1 \oplus w_2 \in W, \forall w_1, w_2 \in W$
- *Asszociatív:*  $(w_1 \oplus w_2) \oplus w_3 = w_1 \oplus (w_2 \oplus w_3), \forall w_1, w_2, w_3 \in W$
- *Kommutatív:*  $w_1 \oplus w_2 = w_2 \oplus w_1, \forall w_1, w_2 \in W$
- $\preceq$  teljes rendezés  $W$ -n:
  - *Reflexív:*  $w \preceq w, \forall w \in W$
  - *Anti-szimmetrikus:* Ha  $w_1 \preceq w_2$  és  $w_2 \preceq w_1$ , akkor  $w_1 = w_2, \forall w_1, w_2 \in W$
  - *Tranzitív:* Ha  $w_1 \preceq w_2$  és  $w_2 \preceq w_3$ , akkor  $w_1 \preceq w_3 \forall w_1, w_2, w_3 \in W$
  - *Teljes:*  $\forall w_1, w_2 \in W: w_1 \preceq w_2$  vagy  $w_2 \preceq w_1$
- $\phi$  összeegyeztethető a  $(W, \oplus)$  Abel-csoporttal  $\preceq$  szerint:
  - *Elnyelés:*  $w \oplus \phi = \phi, \forall w \in W$
  - *Maximalitás:*  $\phi \not\preceq w, \forall w \in W$

**1.2.1. Megjegyzés.** Kiemelném az összehasonlítás operátor ( $\preceq$ ) teljes rendezési tulajdon-ságát, mert az a tulajdonság teszi lineárissá<sup>11</sup> a rendezést.

**1.2.2. Megjegyzés.** A  $\phi$  végtelen elem pusztá létét, ill. összeegyeztethetőségét az Abel-csoporttal szintén fontos kiemelni, hiszen ezáltal szinten bármilyen alaphalmazt megadhatunk az élek bármelyik, tulajdonság-vektorbeli dimenziójának. Emellett érdemes megjegyezni, hogy azon (részben)rendezett halmazok, melyek bármely kételemű részhalmazának létezik infimuma és szuprimuma, hálóknak nevezük. A routing algebrák esetében az természetesen csak akkor teljesül, ha alkalmasan választjuk meg a rendező ( $\preceq$ ) operátorát és az alaphalmazt: a valós számhalmazon, a „hagyományos”  $\leq$  rendezés esetén a routing algebrák hálók.

Most már meg tudjuk mondani egy egyszerű él súlyát. Az útvonalválasztás során azonban nem éleket akarunk összehasonlítani, hanem útvonal-alternatívákat:

**1.2.4. Definíció (Egy út súlya).** Egy  $p = (v_1, v_2, \dots, v_k)$  út  $w(p)$  hosszát az út éleinek súlyainak  $\oplus$ -szerinti összege adja:

$$w(p) = \bigoplus_{i=1}^{k-1} w(v_i, v_{i+1}).$$

Ezután azt mondjuk, hogy egy preferált (tetsző, legjobb, stb.) út az  $\mathcal{A}$  algebrában  $s$  és  $t$  között a legkisebb súlyú  $\preceq$  szerint:

$$Pol(\mathcal{P}_{st}) = p^*: w(p^*) \preceq w(p), \forall p \in \mathcal{P}_{st}.$$

Ezek után könnyen ellenőrizhető, hogy a legáltalánosabban használt routing policy, a shortest path routing (legrövidebb utak) algebrája a  $(\mathbb{R}^+, \infty, +, \leq)$ , míg egy másik policy, a widest-path routing (legszélesebb utak) algebrája a  $(\mathbb{R}^+, 0, \min, \geq)$ .

---

<sup>11</sup>Az első három tulajdonság miatt csak részbenrendezett halmazról beszélhetünk, ha azonban minden elem összehasonlítható, akkor válik teljes, vagy lineáris rendezéssé  $\preceq$ .

### 1.3. Routing algebrák tulajdonságai

A routing algebráknak, mint matematikai struktúrának számos érdekes tulajdonsága van. Ezek közül vannak olyanok, melyek alapvetően befolyásolják az algebra felhasználhatóságát, hiszen az algebra olyan minőségbeli tulajdonságát határozzák meg, mint például az útvonalválasztás algoritmikus lépésszáma. Vannak pusztán leíró jellegű tulajdonságok is, melyek segítenek az algebrák összehasonlításában, az egyes policy-primitívek kiválasztásában is.

#### 1.3.1. Monotonitás, izotonitás és egyéb tulajdonságok

A legtöbb esetben két meghatározó tulajdonsággal kell rendelkeznie egy algebrának, hogy „jól viselkedőnek” mondjuk. Az ilyen algebrákat reguláris algebrának nevezzük:

**1.3.1. Definíció (Reguláris algebra).** Az  $\mathcal{A}$  routing algebrát regulárisnak nevezzük, ha

- Monoton (M):  $w_1 \preceq w_1 \oplus w_2, \forall w_1, w_2 \in W$
- Izoton (I):  $w_1 \preceq w_2 \Rightarrow w_3 \oplus w_1 \preceq w_3 \oplus w_2, \forall w_1, w_2, w_3 \in W$

A monotonitás (M) azt jelenti, hogy egy  $w_1$  súlyú élet, elé illesztve egy  $w_2$  súlyú másik éssel csak kevésbé preferáltható teheti:  $w_1 \preceq w_2 \oplus w_1$ . A kommutativitás miatt igaz az él után való illesztésre is:  $w_1 \preceq w_1 \oplus w_2$ . A hagyományos értelemben vett hosszúságot általánosítja ez a tulajdonság, azaz a „a hosszabb út rosszabb”<sup>12</sup>.

Az izotonitás (I) azt jelenti, hogy adott  $\preceq$  relációban álló éleket ugyanazzal az éssel előlről, vagy hátulról meghosszabbítunk, akkor a relációs viszony nem változik (Az itt leírtak igazak élek helyett utakra is).

Az F.2. függelékben megtalálható a routing teljes modellje, melyet a routing algebrák tervezésekor használtak [20]. Ez a modell a lokális memóriaigényre koncentrál, és azt mondja, hogy egy  $\mathcal{A}$  routing algebra tömöríthetetlen, ha az  $\mathcal{M}_{\mathcal{A}}$  lokális memóriaigény  $\Omega(n)$ , különben  $\mathcal{A}$  tömöríthető. Egy tömöríthetetlen routing algebra nyilván nem skálázódik jól, viszont a tömöríthető algebrák igen. A reguláris algebrák „jól viselkednek”, hisz a monotonitás és izotonitás garantálja, hogy a preferált út megkapható polinom időben az általánosított Dijkstra algoritmussal. Ez lehetővé teszi, hogy legfeljebb  $\tilde{O}(n)$  bit információt tároljunk lokálisan, így nem csak az elméletben, hanem a valóságban is használható algebrákat kaphatunk [12, 20].

A teljesség igénye nélkül felsorolok még néhány tulajdonságot [13]:

- *Delimitált (D):*  $w_1 \oplus w_2 \neq \phi, \forall w_1, w_2 \in W$
- *Szigorúan monoton (SM):*  $w_1 \prec w_1 \oplus w_2, \forall w_1, w_2 \in W$
- *Kiválasztó (S):*  $w_1 \oplus w_2 \in \{w_1, w_2\}, \forall w_1, w_2 \in W$

---

<sup>12</sup>Pontosabban a hosszabb út nem jobb, mint a rövidebb.

- *Elnyelő (C):*  $w_1 \oplus w_2 = w_1 \oplus w_3, \forall w_1, w_2, w_3 \in W$

A fentiek közül talán csak a delimitáltság (D) szorul magyarázatra. Ez a tulajdonság garantálja, hogy az éleket bármilyen önkényesen választott sorrendben is kombinálva járható utat kapunk. Általában az intra-domain<sup>13</sup> routing policy-k rendelkeznek ezzel a tulajdonssággal, de vegyük csak példának a BGP völgymentességét: ha egy csomag a hierarchiában fentebbi AS-től érkezik, akkor azt már csak lefelé mutató, vagy peer kapcsolaton keresztül lehet továbbítani.

#### 1.4. Műveletek routing algebrák között

A routing algebrák megadási módja igen változatos policy-k leírására ad lehetőséget, de – annak érdekében, hogy teljes rendezés lehessen  $\preceq$  – minden csak egy metrika szerint tudunk optimalizálni. Ha szeretnénk több szempontot is figyelembe venni, amire van is példa a routing policy-k között, akkor ezt az algebrák egymásutáni alkalmazásával tudjuk megtenni, így néhány egyszerű policy algebrájával meglepően bonyolult és a valóságot nagyon jól leíró algebrákat tudunk létrehozni [12]. Két ilyen műveletet igen fontos szerepet kap, az egyik a lexikografikus szorzat, ami az összeillesztő-, és a subalgebra, ami a szétválasztó operátor [13].

**1.4.1. Definíció ( $\mathcal{A} \times \mathcal{B}$ ).** Legyenek adottak az  $\mathcal{A} = (W_{\mathcal{A}}, \phi_{\mathcal{A}}, \oplus_{\mathcal{A}}, \preceq_{\mathcal{A}})$  és  $\mathcal{B} = (W_{\mathcal{B}}, \phi_{\mathcal{B}}, \oplus_{\mathcal{B}}, \preceq_{\mathcal{B}})$  algebrák. Ekkor az  $\mathcal{A} \times \mathcal{B} = (W, \phi, \oplus, \preceq)$  lexikografikus szorzatuk, ahol

- $W = W_{\mathcal{A}} \times W_{\mathcal{B}}$ , ill.  $\phi = (\phi_{\mathcal{A}}, \phi_{\mathcal{B}})$
- $(w_1, v_1) \oplus (w_2, v_2) = (w_1 \oplus_{\mathcal{A}} w_2, v_1 \oplus_{\mathcal{B}} v_2), \forall w_1, w_2 \in W_{\mathcal{A}}$  és  $v_1, v_2 \in W_{\mathcal{B}}$
- $(w_1, v_1) \preceq (w_2, v_2) = \begin{cases} v_1 \preceq_{\mathcal{B}} v_2 & \text{ha } w_1 =_{\mathcal{A}} w_2 \\ w_1 \preceq_{\mathcal{A}} w_2 & \text{különben} \end{cases}$

**1.4.1. Megjegyzés.** A  $\phi$  jól definiált, ha  $\mathcal{A}$  és  $\mathcal{B}$  delimitáltak. Egyéb esetben  $\phi$  megadása nem magától értetődő.

**1.4.2. Megjegyzés.** A lexikografikus szorzat nem kommutatív:  $\mathcal{A} \times \mathcal{B} \neq \mathcal{B} \times \mathcal{A}$ .

A korábban már említett shortest-path és a widest-path policy-k algebráinak lexikografikus szorzata az  $\mathcal{SW}$  (shortest-widest – legrövidebb-legszélesebb routing: a legszélesebb útra irányít, ám ha több ilyen is van, akkor azok közül a legrövidebben) és a  $\mathcal{WS}$  (widest-shortest – legszélesebb-legrövidebb routing: először a legrövidebb útra irányít, ha több ilyen is van, akkor azok közül a legszélesebbben).

A második definiált művelet a subalgebra:

**1.4.2. Definíció ( $\hat{\mathcal{A}}$ ).** Adott az  $\mathcal{A} = (W, \phi, \oplus, \preceq)$  algebra és a súlyok egy  $\hat{W} \subseteq W$  részhalmaza. Az  $\mathcal{A}$  algebra leszűkítése  $\hat{W}$ -re:  $\hat{\mathcal{A}} = (\hat{W}, \phi, \oplus, \preceq)$  akkor és csak akkor subalgebrája  $\mathcal{A}$ -nak, ha  $\hat{W}$  zárt  $\oplus$ -ra.

---

<sup>13</sup>Intra-domain: AS-n belüli.

### 1.4.1. A műveletek hatása a tulajdonságokra

A routing algebrák kompozíciójaként létrejött új algebrák tulajdonságai levezethetők az alkotó algebrákéból.

A következő tétel mutatja a lexikografikus szorzat hatását a tulajdonságokra:

#### 1.4.1. Tétel (A lexikografikus szorzat hatása az algebrák tulajdonságaira.). .

- $M(\mathcal{A} \times \mathcal{B}) \Leftrightarrow SM(\mathcal{A}) \vee (M(\mathcal{A}) \wedge M(\mathcal{B}))$
- $I(\mathcal{A} \times \mathcal{B}) \Leftrightarrow I(\mathcal{A}) \wedge I(\mathcal{B}) \wedge (N(\mathcal{A}) \vee C(\mathcal{B}))$
- $SM(\mathcal{A} \times \mathcal{B}) \Leftrightarrow SM(\mathcal{A}) \vee (M(\mathcal{A}) \wedge SM(\mathcal{B}))$

Az (1.4.1) tétel szerint annak, hogy reguláris algebrákat hozunk létre, szükséges feltétele, hogy csakis izonton algebrákat használunk fel, sőt a monotonitás is megkövetelt minden két tagra vagy az első tag szigorú monotonitása.

### 1.5. Példák algebráakra és kombinálásukra

Az 1.5.1 táblázatban néhány példát látunk a leginkább kutatott intra-domain routing policy-kra az algebráikkal és a legfontosabb tulajdonságaikkal. Az utolsót leszámítva az összes felsorolt algebra delimitált és reguláris ( $D, M, I$ ).

Policy	Algebra	Tulajdonság
Legrövidebb út	$\mathcal{S} = (\mathbb{R}^+, \infty, +, \leq)$	SM, I
Legszélesebb út	$\mathcal{W} = (\mathbb{R}^+, \infty, \min, \geq)$	S, I, M
Legmegbízhatóbb út	$\mathcal{R} = ((0, 1], 0, *, \geq)$	SM, I
Legszélesebb-legrövidebb út	$\mathcal{WS} = \mathcal{S} \times \mathcal{W}$	SM, I
Legrövidebb-legszélesebb út	$\mathcal{SW} = \mathcal{W} \times \mathcal{S}$	SM, $\neg I$

1.5.1. táblázat. Routing policy-k, algebráik és fontosabb tulajdonságaik.

A  $\mathcal{W}$  a widest-path routing algebrát jelenti [24]. Ennél az algebránál egy él súlya a kapacitása adja, és nyilván egy több élből álló úton az út kapacitása megegyezik az útmenti legkisebb kapacitással. Emellett nyilván minél nagyobb egy út globális kapacitása, annál inkább preferált. A  $\mathcal{W}$  algebra kiválasztó is és megadható a  $(\mathbb{R}^+, \infty, \min, \geq)$  négyessel (az (F.3.1) tétel értelmében pedig tömöríthető is).

A legmegbízhatóbb út algebra ( $\mathcal{R}$ ) úgy értelmezhetjük, hogy az élek súlyát az a valószínűség adja, hogy az adott élen a csomag hibátlanul át tud menni, így nagyobb érték a jobb. Nyilvánvaló, hogy  $\mathcal{R}$  tartalmaz egy szigorúan monoton (SM) subalgebrát: ha nem engedjük meg a biztos valószínűséget, akkor minden út biztosan romlik, ha hozzáveszünk egy újabb élet, azaz az értéke csökken, mert egy 1-nél kisebb számmal szorozzuk az eddig súlyt:  $\hat{\mathcal{R}} = ((0, 1], 0, *, \geq)$ .

A legszélesebb-legrövidebb út (widest-shortest path)  $\mathcal{WS}$  routing esetén a legrövidebb utak közül a legnagyobb kapacitású választjuk [1], míg a legrövidebb-legszélesebb út (shortest-widest path)  $\mathcal{SW}$  ([16, 24]) routing esetén, éppen fordítva, a legnagyobb kapacitású utak közül a legrövidebbet választjuk. Ezeket az algebrákat megkaphatjuk a  $\mathcal{S}$  és a  $\mathcal{W}$  algebrák lexikografikus szorzataiként [13].

**1.5.1. Megjegyzés.**  $\mathcal{SW}$  nem izoton. Az (F.3.2) tételet áll a nem izoton algebráakra is, így  $\Omega(n)$  bit lokális memóriát igényel a  $\mathcal{SW}$  algebra is. Jelenleg még nyitott kérdés, hogy ez a határ szoros-e. Nem tudjuk, hogy van-e jobb megoldás, mint a triviális, azaz hogy minden router tárol egy-egy routing tábla bejegyzést minden forrás-cél párra, ami  $O(n^2 \log d)$  ( $d$  a maximális fokszám) bit per router memóriaigényű.

## 1.6. Összefoglaló

Ebben a fejezetben áttekintettem a szakirodalmat, összeszedve a legfontosabb állomásokat. Az 1990-es évek végétől egyre többet foglalkoztak a kutatók a BGP hálózatának feltárással, amivel el is jutottak addig, hogy van egy viszonylag pontos kép a hálózatról, de arra, hogy miért egy adott policy-t használnak az AS-ek, nem tudnak válaszolni. Rámutattam, hogy az Internet AS-szintű topológiáján kívül, a más tudományterületekről származó problémák útvonalválasztásáról sem tudunk sok minden és ezért van szükség egy olyan eszközre, ami a policy-feltáras feladatát – általános esetben is – hatékonyan el tudja láttni. Ehhez definiáltam a routing algebrát, bemutattam a legfontosabb tulajdonságait és műveleteit, emellett a legszélesebb körben használt policy-k algebráit is ismertettem.

## 2. fejezet

# A hálózatkutatás legfontosabb modelljei

Számos kutatás foglalkozik különböző hálózatokon terjedő információk vizsgálatával. Természetesen minden ilyen modellt le lehet írni az általanosított hálózati modellel, így a meghatározó policy-k azonosítása után vizsgálható az 1. fejezetben bemutatott, vagy azokhoz hasonló algebrákkal. Ebben a fejezetben bemutatom a hálózatkutatás leggyakrabban vizsgált modelljeit, és leírom a modellekhez tartozó meghatározó policy-ket is.

Az útvonalválasztás szempontjából alapvető különbség van a lokális- és a globális optimalizálás között. Míg globális optimalizálás esetén az egész hálózatot figyelembe véve választjuk meg a legjobb útvonalat, addig lokális optimalizálnál csak a helyi viszonyok számítanak. Jelenleg a számítógépes hálózatokban a két legelterjedtebb használt policy kategória a „distance-vector”<sup>1</sup> és a „link-state”<sup>2</sup>. A módszerek abban különböznek, hogy a hálózatról milyen információkat gyűjtenek és hogyan, de abban megegyeznek, hogy globális optimumra törekszenek, tehát pl. a legrövidebb utat keresik meg. Ezzel szemben pl. a greedy routing<sup>3</sup> csupán törekszik elérni globális optimumot, ám könnyen elakadhat, ha nem javítunk az alapalgoritmuson [3]. Könnyen belátható, hogy a vírusterjedés-jellegű feladatok lokális optimumra törekszenek, míg az Internetes útvonalválasztás globálisan optimalizál.

A másik szempont, ami meghatározó, az az, hogy a csomópontok felülről tudják-e bírálni a csomagok terjedését abban az értelemben, hogy a küldő eredeti célját meg tudja-e valósítani akkor is, ha az útvonal mentén valamelyik csomópont ezt nem akarja. Feltehetjük-e, hogy a hibamentes csomagtovábbítás elsődleges prioritás minden hálózati csomópontnak (pl. kölcsönös bizalom megteremtése érdekében), vagy az egyes csomópontok viselkedhetnek önkényesen?

Mivel ezen két tulajdonság nem zárja ki egymást, a legtöbbször azzal az esettel állunk szemben, hogy nem tudunk egy problémát egyértelműen karakterizálni.

---

<sup>1</sup>Távolság-vektor módszerek, a Bellman-Ford algoritmusra épül.

<sup>2</sup>Link-állapot módszerek, a Dijkstra algoritmusra épül.

<sup>3</sup>Mohó útvonalválasztás: A lokális helyzet optimumát választja.

## 2.1. Vírusterjedés komplex hálózatokban

A vírusok terjedését és járvánnyá fejlődését sok évtizede vizsgálják. A járványok előrejelzése lehetőséget ad a tudósoknak, hogy megtervezzék a védőoltások ütemezését és az esetleges karanténok felállítását, ami jelentős hatással lehet egy adott betegség halálozási rátájára. A fertőző betegségek modellezése egy olyan eszköz, ami segítséget ad a fertőzés terjedésének vizsgálatához és a jövőbeli járványok elkerüléséhez stratégiákat lehet kidolgozni általa [6].

### 2.1.1. A vírusterjedés matematikai modellje

A legkorábbi ismert matematikai modell a fertőző betegségek terjedéséről Bernoullitól származik, aki a fekete himlőt vizsgálva megállapította, hogy ha mindenkit beoltanának a betegség ellen, akkor az átlagos életkor 26 év 7 hónapról 29 év 9 hónapra nőne [2].

Bernoulli munkája során természetesen még nem értette annyira a baktériumok és vírusok biolóját, mint mi. A XX. század első felében Ronald Ross malária kutatásával kezdetét vette a modern elméleti fertőzés kutatás. Ezután nem sokkal, 1927-ben A. G. McKendrick and W. O. Kermack elismert munkája is megjelent, az „A Contribution to the Mathematical Theory of Epidemics”. Ez a determinisztikus modell sikeresen jelezte előre sok fertőző betegség viselkedését.

Amikor nagy populációkat vizsgálunk, a determinisztikus modelleket használjuk, mert egy ilyenben a populáció egyedeit besorolhatjuk alcsoportokba, ami a fertőzés egy specifikus stádiumát jelenti. Ha a fertőzés átterjedése az egyik csoportról a másikra időben folytonos, a csoportok aktuális mérete matematikailag a deriválással fejezhető ki, így a modell leírható differenciál egyenletekkel. Ott játszik fontos szerepet a determinisztikusság, hogy egy ilyen modellben feltesszük, hogy egy csoport egyedszáma deriválható az idő szerint, ehhez viszont az kell, hogy a fertőződés kiszámíthatóan (nem véletlenszerűen) terjedjen tovább, más szóval a csoportok egyedszámának változása egyértelműen meghatározható az addigi múltból. Ezen modellek szakirodalmát és terminológiáját áttanulmányozva számos különböző mérőszámot és tulajdonságot találhatunk, de az alapmodell a következő három csoportot jelöli:

- $S(t)$ : A  $t$  időpontig meg nem fertőzött egyedek száma, azaz akik még megfertőződhetnek.
- $I(t)$ : A  $t$  időpontig megfertőzött egyedek száma, azaz akik tovább tudnak fertőzni.
- $R(t)$ : A  $t$  időpontig megfertőzött, de mégsem  $I(t)$ -beli egyedek száma, azaz vagy meggyógyultak, vagy meghaltak. Ezen csoport egyedei már nem tudnak újra megbegegni, sőt fertőzni sem fertőznek.

Egy egyed az  $S(t) \rightarrow I(t) \rightarrow R(t)$  sorrendben halad a csoportok között. Egy fix méretű

populációt feltételezve,  $N = S + I + R$ , a következő egyenleteket vezethetjük be [14]:

$$\frac{dS}{dt} = -\beta SI \quad (2.1)$$

$$\frac{dI}{dt} = \beta SI - \gamma I \quad (2.2)$$

$$\frac{dR}{dt} = \gamma I \quad (2.3)$$

Ahol  $\beta$  a kapcsolati ráta és  $1/\gamma$  az átlagos fertőző periódus.

Néhány feltételezést tettünk az egyenletek felírásához: (1) minden egyedet ugyanakkora valószínűséggel fertőz meg egy  $\beta$  kapcsolati rátájú betegség, így minden fertőzött egyed  $\beta N$  egészséges egyedet fertőz meg egy időegység alatt, illetve az egyedeknek átlagosan  $S/N$  megfertőzhető kapcsolata van. Tehát a fertőzési ráta: az új betegek száma  $\beta N(S/N)$ , azaz amennyivel változik (nyilván csökken)  $S$ :  $\beta N(S/N)I = \beta SI$ .

Ezután érthető a második és harmadik egyenlet: a fertőzöttek száma annyival nő, ahánnyal kevesebb egészséges lesz, illetve annyival fogy, ahány meghal vagy meggyógyul. A  $\gamma$  jelöli az átlagos halálozási / felépülési rátát.

Végül feltessük, hogy a megfertőződés és a felépülés időben jóval gyorsabb folyamat, mint a születés és halálozás, így ezeket a faktorokat kihagyhatjuk a modellből.

**2.1.1. Megjegyzés.** Természetesen számos következő szempontot figyelembe lehet még venni a modell mélyítése érdekében, ilyen például az előbb említett születés és halálozás; az  $R \rightarrow S$  átmenet, azaz a fertőzésből való meggyógyulás után újra meg lehet betegedni; figyelembe lehet venni az úgynévezett lappangási időszakot, bevezetve egy újabb csoportot ( $S \rightarrow \epsilon \rightarrow I \rightarrow R$ ); vagy a veleszületett betegséget is, azaz a fertőzött anyától elkapott betegséget is.

Jelen dolgozat szempontjából az alap SIR modell megfelelő.

### 2.1.2. Vírusterjedés, mint útvonalválasztási probléma

A fertőző betegségek terjedése nem látszik útvonalválasztási kérdésnek sőt, semmilyen policy-t nem veszünk észre első ránézésre, ezért talán némi magyarázatot igényel a problémakör vizsgálata. A diplomadolgozat végső célja rejtett szabályok felismerése előre definiált policy primitívek segítségével tetszőleges hálózatban, így világos, hogy a vírusterjedést sem zárhatom ki a vizsgálatból.

A fejezet bevezetőjében említett két tulajdonság közül az optimalizálás az, ami itt a legszembetűnőbb. Ha ugyanis nem úgy tekintünk a problémára, mint egy kétszereplős „háború” (ember a vírus ellen), hanem úgy, mint a kis organizmusok egyéni útvonalválasztására, akkor látható, hogy lokális optimalizálásról van szó, hiszen minden egyes fertőző elem szeretne minél több előre meghatározott célponton keresztül elérni a célát. Lévén itt nem a tipikus több lépésen keresztfüggő pont-pont routing-ról van szó, globális optimalizálási feladat legfeljebb az lehet, hogy minél több csomópont legyen fertőzött. minden csomópontból arra fertőz tovább a vírus, amerre a legtöbb esélytel fog továbbélni – ez nyilván

függ az adott vírus-egyedtől is -, azaz például a leggyengébb immunrendszerű ember felé. Ez alapján két policy-t mutatok be:

- Fertőzési-határ: A vírus-egyed legjobb továbbfertőzési policy-ja.
- Unió-fedés: A globális fertőzés policy-ja.

### Fertőzési-határ

A Fertőzési-határ policy alapú útvonalválasztás garantálja a legjobb túlélési esélyt egy vírus-egyed számára. Ennek a policy-nak az algebrája az  $\mathcal{F} ((0, 1], 0, \max, \geq)$ . Ennél az algebránál az élek súlya (egy  $(0, 1]$  intervallumbeli valós szám) azt adja meg, hogy azt a csomópontot, ahová vezet, milyen valószínűsséggel tudjuk megfertőzni, így nyilván a nagyobb jobb, a 0 pedig azt jelenti, hogy a csomópont  $R(t)$ -beli, azaz már biztosan nem lehet megfertőzni. Mivel az útvonalválasztás lokálisan optimalizál, ezért két alternatív élet a  $\geq$  operátorral tudunk összehasonlítani, illetve a két él összeillesztésére a két él súlyának maximumát vesszük.

### Unió-fedés

Az Unió-fedés policy szerinti útvonalválasztás célja, hogy lehetőleg olyan élen menjünk és fertőzzünk tovább, amit még nem használtak. Ennek a policy-nak a számítógépes vírusok tárgyalásánál van nagyobb szerepe, semmint a biológiai fertőzéseknel, hiszen ez alapján – akár már fertőzött gépet is – minden olyan élet saját felügyelet alá vehetünk, amit még nem találtunk meg, így például ügyelhetünk arra, hogy semelyik fertőzött gép ne tudja frissíteni a vírusirtójának a vírusdefiníciós adatbázisát.

Az Unió-fedés policy algebrája az  $\mathcal{U} (\mathbb{N}, \infty, f, \leq)$ . Itt az élsúly egy természetes szám lehet és azt fejezi ki, hogy hányszor használták már fel egy útvonalválasztásban. Mivel a cél az, hogy minél ritkábban (optimálisan soha nem) használt utakat használunk, a  $\infty$  jelenti az átváratlanosságot és a  $\leq$  az összehasonlító operátor. A több élből álló út súlyát az  $f$  összegző függvény adja:

$$f(e_1, e_2) = \begin{cases} e_1 + e_2 & \text{ha } e_1 * e_2 \neq 0 \\ 0 & \text{különben} \end{cases}$$

Ezzel garantáljuk, hogy olyan esetben, amikor úgyis mindegy – azaz nincs még fel nem használt él – akkor az élek súlyának összegét vesszük, de amikor van olyan él, amit még nem használtunk, akkor azt biztosan fogjuk használni (hacsak nincs egy másik alternatív 0 súlyú út), hiszen a két élű út súlya 0 lesz.

## **2.2. Trendterjedés közösségi hálózatokban**

A közösségi hálózatok az elmúlt 10 év egyik legmeghatározóbb jelensége, új korszakot nyitott az emberi kapcsolatokban. A kapcsolatháló elemzés a szociometriából indult, de annál bonyolultabb, időben is változó, nagyobb közösségek vizsgálatára alkalmas módszer-tan. A kapcsolatháló elemzési módszerek alkalmasak egy nagyobb csoport klickjeinek és alcsoporthajnak felrajzolására és megjelenítésére is. A kapcsolathálózati megközelítés kevésbé hangsúlyozza az egyéni cselekvés szerepét a struktúrák létrehozásában, nem az egyéni szándékból, hanem a struktúrák belső feszültségeiből jön létre a cselekvés mozgástere. A struktúra – pl. egy elterjed trend – e szemléletmódban számára nem egy közvetlenül megmutatkozó adottság, hanem sokkal inkább a kapcsolatok hálójából bonyolultan kibontható szerveződés.

A hálózati megközelítés előnye, hogy az adott hálózati struktúrákat, közösségeken belül kialakuló attribútumokat dinamikus módon, a változásokon keresztül is képes vizsgálni: a kapcsolatok és a környezet folytonosan változó mozgása mentén. A hálózat elemzéssel komplex módon tudjuk leírni vizsgált közösségi hálózatok működését [5].

A trend- vagy divatterjedés egy szociális hálóban lehetséges, ahol az egyik csomóponttal szimbolizált egyén felvesz egy szokást, elkezd hordani egy bizonyos ruhadarabot (vagy márkát), vagy bármi olyat tesz, amit „az átlag” nem. Ha az ismerősei – a kapcsolati hálóban szomszédos csomópontok – észreveszik ezt az újdonságot és megtetszik nekik is, átveszik a forrástól. A problémakör alapvető kérdése, hogy mi határozza meg, hogy elterjed-e egy trend és divattá válik-e az egész hálózatban, vagy sem. Malcolm T. Gladwell brit-kanadai újságíró, író 2000-ben megjelent könyvében [10] számos példát olvashatunk, hogy egyes szituációkban mikor érkezett el a fordulópont, mikor vált általános divattá valami. Ilyen példa a Hush Puppy cipőgyártó cég, mely egyik pillanatról a másikra a csőd széléről világ-szerte ismert márka vált. Gladwell azt állítja, hogy a kulcs az úgynevezett *Összekötőkön* műlik. Egy összekötő tipikusan hatalmas társasági kapcsolatrendszerrel rendelkezik, mely kapcsolatok többnyire gyenge kötelék csupán, emellett több mikrovilágba és szubkultúrába bejáratos. Az ilyen összekötőket kell megnyernie egy terméknek és ennek szinte automatikus következménye, hogy elterjed a termék és divattá válik. Duncan Watts ausztrál fizikus-matematikus-szociológus megkérdőjelezte Gladwell állítását, szerinte ugyanis nem az a meghatározó, hogy egy divat az összekötőktől indul-e, hanem az, hogy a közösség (akár az egész társadalom, akár csak egy szűkebb baráti kör) készen áll-e az adott divat elterjedésére. Tehát nem a forrás számít és annak befolyása, hanem az, hogy általánosságban a közösség egésze akarja-e a terméket.

### **2.2.1. Trendterjedés, mint útvonalválasztási probléma**

A trendterjedés nagyon hasonlít a vírusterjedésre, hiszen maga a két eredeti modell is szinte egy az egyben megfeleltethető egymásnak. Természetesen a vírusterjedés esetén említett kiegészítéseket nem lehet egy az egyben átültetni és a trendterjedés terminológiájával le-

írni, de az alapmodell három csoportját ( $S$ ,  $I$ ,  $R$ ) itt is azonosíthatjuk. Világos, hogy az optimalizálás lokalizáltsága erre a modellre is jellemző, hiszen nincs olyan divat, ami egy bizonyos emberre akar csak hatni, sokkal inkább mindenki. Így ennek a problémakörnek is jellemző policy-ja lehet az *Unió-fedés*.

Útvonalválasztási szempontból a lényeges különbség, hogy a trendterjedésnél a csomópontoknak van döntő szerepe, nem az éleknek, azaz itt nem közös érdek, hogy egy divat elterjedjen (vagy akar az, hogy ne terjedjen el). Bárki mondhatja azt, hogy neki kifejezetten nem tetszik az elterjedő divat, ezért pl. lebeszéli az ismerőseit róla, vagy csak egyszerűen nem veszi át, így megállítja a terjedést, legalábbis a közvetlen közelében. Nyilvánvaló, hogy – bár az elterjedés kulcsa az, hogy a társadalom készzen áll-e – egy *Összekötő* nagyban hozzájárulhat a sikeres terjesztéshez, így preferálhatjuk ezen központi csomópontokat. Emellett az is szempont lehet, hogy nem a csomópontok befolyását vesszük alapul, hanem pusztán a mennyiségiüket és egy adott útvonalválasztáskor minél hosszabb útra törekszünk. Egy másik alkalmazható heurisztika, hogy ha két csomópont közötti élen sok divat terjedt át, akkor ezt az élet érdemes felhasználni, illetve megfordítva, ha egy élen eddig kevés divat terjedt át, akkor valószínűleg nem olyan a két végpont kapcsolata, amin a divatok átterjedhetnek, így érdemes elkerülni az ilyen éleket. Ezek alapján a következő policy-kat definiálom:

- **Összekötő-keresés:** A nagy súlyú csomópontok felé irányító policy.
- **Korai-elfogadó-keresés:** A minden átvevő csomópontok felé irányító policy.

### **Összekötő-keresés**

Ez a policy azon csomópontokat részesíti előnyben, amelyeket a legbefolyásosabbnak ítélik. A vírus- és trendterjedés hasonlóságából adódóan itt is lokális optimalizálásról beszélhetünk, így minden azt a szomszédot fogjuk választani, aminek a legnagyobb a fokszáma. Az *Összekötő*-keresés algebrája a  $\mathcal{O}((1, d), 0, \max, \geq)$ . Az élek súlya:  $\forall u, v \in V : w(u, v) = \max(\deg(u), \deg(v))$ , tehát az összekötött csomópontok fokszámai közül a nagyobb és nyilván a több jobb. Az átjárhatatlanságot a 0 súly jelzi, míg több él súlyát a maximumuk határozza meg.

### **Korai-elfogadó-keresés**

A Korai-elfogadó-keresés policy azon éleket részesíti előnyben, melyeken az eddigi tapasztalat alapján sok divat átterjedt már. Ennek lényege nyilván az, hogy egy ilyen él olyan személyek közötti kapcsolatot jelöl, akik pl. megbíznak a másik ítéletében, vagy az egyikük felnéz a másikra, így átvesz tőle viselkedési formákat. A policy algebrája a  $\mathcal{K}(\mathbb{N}, -1, +, \geq)$ . Az élek súlya azt jelzi, hogy eddig mennyi divat terjedt át rajta, így a több a jobb, és nyilván a -1 súlyú élen nem tudunk átmenni. Több él összeillesztését az élsúlyok összegével súlyozunk.

### **2.3. Útvonalválasztás az Interneten**

A útvonalválasztás kutatói rendszerint informatikai-, vagy számításelméleti szakemberek, ezért a kutatási célok legnagyobb része számítógépes hálózatok útvonalválasztásának vizsgálata, javítása. Ezen a területen általában az Internet gerinchálózatát szokták vizsgálni, hiszen ott jelentkeznek azok a skálázhatósági- és menedzsmentproblémák, amik az Internet 1970-es évekbeli tervezéséből származnak és ami miatt azt mondhatjuk, hogy az Internet hibáinak toldozása-foldozása már nem elég és alapvetően új megoldások szükségesek.

Két alapvetően eltérő koncepció létezik a számítógépes hálózatok útvonalválasztásában, a tapasztalatok alapján előre becsült forgalmi viszonyoknak megfelelő centralizált kialakítás; illetve az aktuális forgalmi helyzet állandó figyelése alapján, az annak legjobban megfelelő irányítás. Legtöbbször az utóbbit valósítják meg, méghozzá elosztott változatban. Az Internet tervezése során több olyan szempontot is figyelembe vettek, melyeket az előző két alfejezetben nem tudunk tárgyalni. Ilyen például a routerekben lévő útvonaltáblák minél kisebb mérete (kisebb tár, olcsóbb és gyorsabban működő csomópont, illetve kisebb routingforgalom), emellett fontos szempont a routingforgalom minimalizálása, a robusz-tusság (hibás tábla esélyének minimalizálása: „fekete lyuk”, hurok, oszcilláció elkerülése) és végül az optimális útvonalak kijelölése (természetesen az út optimalitása az igénytől függ). Az egyik legfontosabb igény az elosztott működés, a decentralizáltság volt azért, hogy az útvonalválasztást a csomópontok végezzék.

Ahogy fejlődött a világ, újabb feladatok és problémák jelentek meg, amelyet a szakemberek a hálózat tervezésekor még nem láttak előre. Olyan új technológiák jelentek meg, mint a mobiltelefonok és a hordozható számítógépek és velük együtt az igény a mobil routingra, azaz arra, hogy ne csak a számítógép, de az internet hozzáférés is legyen hordozható. Problémát jelent a multicast routing is, amikor egy adott csomagot több címvettethez szeretnénk eljuttatni. Ezen problémák hatékony megoldása kritikus feladat, alapvetően új megoldásokat kívánnak.

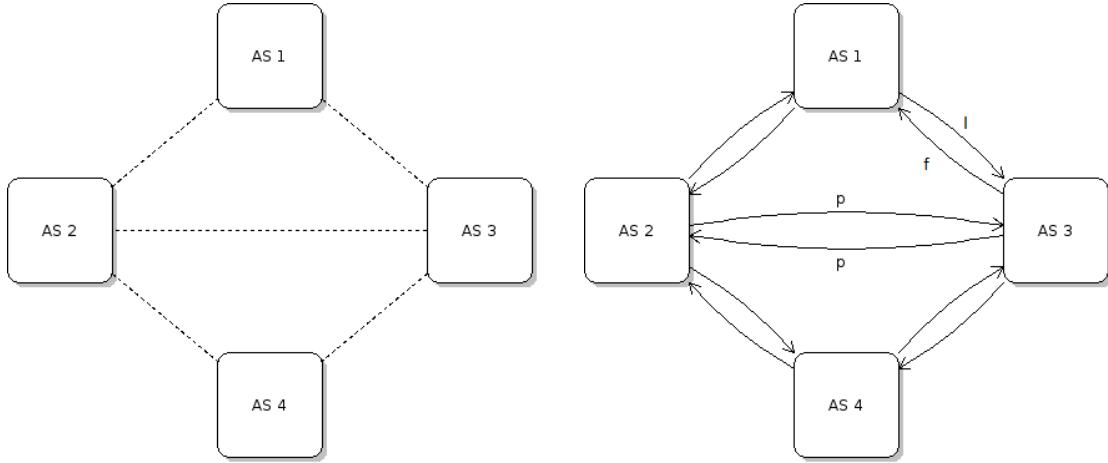
Emellett pusztán a hálózat mérete is feszegeti az Internet teljesítőképességének jelenlegi határait. Mára az Internet óriásira nőtt és a 1990-es évek vége óta a minden nap élet meghatározó részévé vált. 2000 és 2009 között 394 millióról 1,84 milliárd nőtt az Internet kapcsolattal rendelkezők száma<sup>4</sup>. Ma már több, mint 2,4 milliárd<sup>5</sup> felhasználója van az Internetnek, és 2010-ben 12,5 milliárd Internetre csatlakoztatott eszköz volt. Ez azt jelenti, hogy minden élő emberre jutott 1,84 készülék. 2020-ra az Internetre kapcsolt eszközök száma elérheti az 50 milliárdot [9].

---

<sup>4</sup>Market Information and Statistics, International Telecommunications Union, <http://www.itu.int/ITU-D/ict/statistics/>

<sup>5</sup>2.405.518.376 – 2012. második negyedév, <http://www.internetworldstats.com/stats.htm> 2014. 05. 09

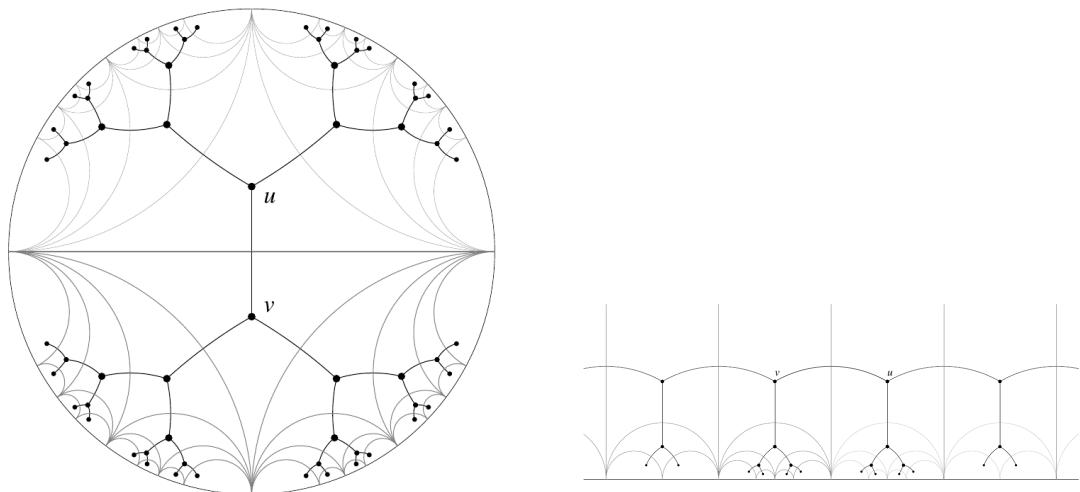
Útvonalválasztási szempontból az Internet routing egy klasszikus feladat, amikor a legszélesebb-, legrövidebb- és legmegbízhatóbb utakat keressük. Ezen policy-k algebráit már bemutattam, megtalálhatók az 1.5.1 táblázatban.



**2.1. ábra.** A BGP egyszerűsített képe és a völgymentesség szerinti irányítás.

Érdemes megvizsgálni a BGP szabályrendszerét: a BGP routing policy-ja többszintű. A legalsó szint, a legalapvetőbb policy a völgymentesség. Ez azt jelenti, hogy az útvonalválasztásnál elsődleges szempont, hogy semelyik AS-nek ne kelljen fizetni olyan forgalomért, ami csak áthalad rajta. Ha a hierarchiában lefelé mutató éleket  $l$ -lel, a felfelé mutató éleket  $f$ -vel, a peering kapcsolatokat pedig  $p$ -vel jelöljük, akkor minden routing során kijelölt útvonal csak a következő alakban írható le: néhány (akár nulla)  $f$  él, aztán maximum egy  $p$  él, utána pedig néhány (akár nulla)  $l$  él.

Emellett, ha felhasználjuk a gráfbeágyazás és kompakt routing eredményeit, akkor egy új távolságfüggvénnyel és a csomópontok koordinátázásával megvalósítható az Internet tartományszintű gráfjában is egy elakadásmentes mohó útvonalválasztás [7].



**2.2. ábra.** A hiperbolikus sík Poincaré-féle diszk modellje és a felső félsík modellje [15].

A  $\mathbb{H}$  hiperbolikus síknak többféle szemléltető modellje is van, ám a két legelterjedtebb a felső félsík modell és a Poincaré diszk modell, amiket a 2.2 ábrán láthatunk is. A Poincaré modellben  $\mathbb{H}$ -t egy egységkörrel reprezentálják:  $x^2 + y^2 < 1$ , a következő metrikával:

$$ds^2 = \frac{4(dx^2 + dy^2)}{(1 - x^2 - y^2)^2}. \quad (2.4)$$

A felső félsík modellben  $\mathbb{H}$ -t a  $\{\langle x, y \rangle \mid y > 0\}$  ponthalmaz írja le, ahol a metrika:

$$ds^2 = \frac{dx^2 + dy^2}{y^2}. \quad (2.5)$$

Mindkét esetben a  $\mathbb{H}$  pontjait komplex számokként kezeljük:  $(x, y) \in \mathbb{R}^2 : z = x + yi$ . Ezek alapján két új policy-t mutatok be:

- Hiperbolikus-távolság: Az elakadásmentes mohó útvonalválasztás policy-ja.
- Völgumentesség: A BGP alapvető, elsődleges policy-ja.

### Hiperbolikus-távolság

A hiperbolikus síkra ágyazott Internet gráf minden pontja egy  $(x, y)$  koordinátapárral leírható. A policy algebrája a  $\mathcal{H}(\mathbb{R}^+, \infty, f_{\mathbb{H}}, \leq)$ , ahol  $f_{\mathbb{H}}$  egy viszonylag bonyolult függvény, definiálásához szükséges lenne több, itt nem részletezett ismeret a hiperbolikus geometria téma köréből (bővebben ld. [22] 2. fejezetét).

### Völgumentesség

A völgumentesség policy-nak az algebrája a  $\mathcal{V}(\{f, l, p\}, \phi, \oplus_{\mathcal{V}}, \preceq)$ , ahol a  $\oplus_{\mathcal{V}}$  a 2.3.1 táblázat szerinti<sup>6</sup>. Az előbbi szabály másik megfogalmazásban azt jelenti, hogy sem  $l$ -t, sem  $p$ -t nem követhet sem  $p$ , sem  $f$ .

**2.3.1. táblázat.** A  $\oplus_{\mathcal{V}}$  [18].

$\oplus$	$f$	$l$	$p$
$f$	$f$	$l$	$p$
$l$	$\phi$	$l$	$\phi$
$p$	$\phi$	$p$	$\phi$

---

<sup>6</sup> Egy adott súlyú (típusú) úthoz hozzá akarnánk venni egy élet, akkor az út minden súlyúvá (típusúvá) válna.

## 2.4. Egyéb algebrák

A minél részletesebb vizsgálat érdekében definiálok olyan policy-kat is, melyek vagy minden eddigi modellben használható lenne, vagy egyikben sem, így eddig nem volt lehetőségem bemutatni.

A fejezet legelején felvázolt két dimenzió (optimalizálás, érdekek) mellett a harmadik karakterizálási lehetőség az időbeli lefolyás. Ha egy útvonalválasztási probléma tárgyalása során figyelembe vesszük az időt is, mint befolyásoló tényezőt, egy olyan új dimenziót ragadunk meg, mely minden modell routing-ját képes befolyásolni: ez az *Időfüggés* policy. A vírus-terjedsnél elég csak arra gondolni, hogy télen könnyebben tud terjedni a fertőzés, és hasonlóan nyáron nagyobb valószínűséggel terjed egy fürdőruhádivat. Fontos azonban, hogy ez csak egy mellékes faktor, azaz egy alap policy mellett van értelme figyelembe venni az időpontot is. Az Internetes útvonalválasztás során időfüggést tapasztalhatunk, ha pl. egy terheléselosztó rendszer egy adott kérést másodpercenként váltakozva egyszer kiszolgál, egyszer pedig újrapróbálkozásra szólít fel.

Másik érdekes policy, a trendterjedsnél említett leghosszabb út policy. Ennek olyan esetben van értelme, amikor nem a célba érkezés a legfontosabb, hanem maga az út. A vírus- és trendterjedsnél ez azért volt lényeges, hogy minél több embert elérjünk, de pl. a hangyák is így építik ki a bolyban az utakat, hogy egy esetleges betolakodó minél nagyobb valószínűséggel eltévedjen.

### Leghosszabb-út

A leghosszabb-út policy adja az elérhető leghosszabb utat. Ebben az esetben az  $\mathcal{L}$  algebra az  $(1, 0, +, \geq)$  négyes. Ebben a policy-ban minden él konstans 1 súlyú, egy út súlya éppen az élszáma.

### Időfüggés

Az Időfüggés policy lényege, hogy az időpontot<sup>7</sup> figyelembe véve néha átjárhatatlan egy-egy él. Ehhez szükséges egy alap policy is, amelyet ez ki tud egészíteni:  $\mathcal{A}$ . Jelölje  $T_{e_i}$  az időpontok egy olyan halmozát, melyben  $\mathcal{I}$  nem enged át forgalmat az  $e_i$  élen. (A  $T_{e_i}$  lehet akár egy  $[t_0, \infty)$  intervallum is.) Így az  $\mathcal{I}$  algebra:  $(W_{\mathcal{A}}, \phi_{\mathcal{A}}, f, \preceq_{\mathcal{A}})$ , ahol

$$f(e_1, e_2, t) = \begin{cases} \phi_{\mathcal{A}} & \text{ha } t \in T_{e_1} \cup T_{e_2} \\ e_1 \oplus_{\mathcal{A}} e_2 & \text{különben} \end{cases}$$

Azaz, alkalmas időpontban teljes egészében az  $\mathcal{A}$  policy érvényesül, azonban minden élen bizonysos időpontokban nem lehet átmenni.

---

<sup>7</sup>A lépték mértéke problémafüggő: lehet naponkénti éves periódusokkal, lehet

## 2.5. Összefoglaló

Ebben a fejezetben megvizsgáltam a hálózatkutatás szempontjából alapvető modelleket, melyeket a lokális- vagy globális optimalizálás és a közös- vagy egyéni érdekek követése tulajdonságok alapján karakterizáltam. Bemutattam a fertőző betegségek vizsgálatára használt matematikai modellt, megvizsgáltam a téma kör útvonalválasztási kérdéseit és ki-jelöltet a két, a problémakört jól jellemző policy-t: a Fertőzési-határ és az Unió-fedés policy-ket, illetve ezek algebráit:  $\mathcal{F} = ((0, 1], 0, \max, \geq)$  és  $\mathcal{U} = (\mathbb{N}, \infty, f, \leq)$ .

Rávilágítottam a trend- és a vírusterjedés hasonlóságaira és különbségeire útvonalválasztási szempontból és definiáltam két policy-t, a Összekötő-keresés-t és a Korai-elfogadó-keresés-t. Ezen policy-k algebrái:  $\mathcal{O} = ((1, d), 0, \max, \geq)$  és  $\mathcal{K} = (\mathbb{N}, -1, +, \geq)$ .

Végül megvizsgáltam a már ismertetett policy-kon (ld. az 1.5. alfejezetet) kívül az Internet tartományszintű gráfjának az alapszabályát, a Völgymentességet, illetve felvázoltam a hiperbolikus térbe ágyazás – általa pedig az elakadásmentes mohó útvonalválasztás – lehetőségét:  $\mathcal{V} = (\{f, l, p\}, \phi, \oplus_{\mathcal{V}}, \preceq)$  és  $\mathcal{H} = (\mathbb{R}^+, \infty, f_{\mathbb{H}}, \leq)$ .

### 3. fejezet

## A modellezési keretrendszer

A 2. fejezetben bemutattam, hogy hogyan lehet az ott ismertetett problémák vizsgálatára hasznosítani a routing algebrák matematikai modelljét. Ahhoz, hogy ezt a gyakorlatban is fel tudjuk használni, az algebrákat szoftveresen is felhasználható módon kell leírni. Emellett ki kell dolgozni azt a modellezési, szimulációs eljárást is, amely keretrendszerként képes a változatos bemeneteket kezelve elemezni a különböző útvonalválasztási szabályrendszerek felderítését.

Ezt a keretrendszert ebben a részben mutatom be. A 3.1. rész elején pontosítom, hogy mit lehet elvárni egy ilyen rendszertől. Részletezem az adatok előzetes feldolgozására, tisztítására vonatkozó elvárásokat, amelynek során kitérek a feladat domainének esetleges szűkítésére is. Kijelölöm a szimulációtól elvárt működést és definiálom, hogy milyen kimenete legyen a rendszernek.

Ezután a 3.2. részben a keretrendszer egy konkrét megvalósítását írom le, amelynek részeként ismertetem az algebrák implementálást, elemzem a felhasznált útvonalválasztó (optimalizáló) algoritmust. Ezután bemutatom, hogy hogyan értékeltem ki a szimulációs eredményeket, hogyan hasonlítom össze az eredetileg választott (megfigyelt) útvonalakat a policy primitívek által adott utakkal.

Végül a 3.4. részben bemutatom, hogy egy valós hálózatot (repülési adatok alapján) hogyan lehet vizsgálni az itt ismertetett keretrendszer segítségével.

### 3.1. Specifikáció

Ahogy a szoftvertermékeknél általában, ebben az esetben is meghatározó, hogy milyen specifikációnak kell megfelelnie a modellezési keretrendszernek. Ehhez természetesen meg kell határozni, hogy milyen célokat szeretnénk elérni és ha erre szükség van, azt is, hogy hogyan szeretnénk céljainkat elérni.

A keretrendszer fő célja, hogy a routing algebrák felhasználásával elemezni és akár visszafejteni is képesek legyünk kívülről megfigyelt útvonalválasztási szabályrendszereket. Ehhez alapvető, hogy a keretrendszer megfelelő rugalmassággal tudja használni az algebrákat, a

definiált algebrák műveleteit képes legyen használni, valamint tudja az algebrákon értelmezett műveleteket is kezelní. Emellett szükség van egy általános formátumra is, amelyre a bemeneti adatokat át tudjuk konvertálni. Az ebben a formátumban lévő adathalmazon értelmezni kell egy optimalizáló függvényt, amely az adott algebrák rendező operátora ( $\preceq$ ) alapján kiválasztja utak egy halmazának leginkább preferált útjait.

### 3.1.1. A formátum

Mivel szeretnénk minél több problémát vizsgálni a rendszer segítségével, pontosítani kell, hogy milyen az elvárt bemenet. Le kell képezni az általános problémát egy jól strukturált, könnyen kezelhető modellre. Ahogy ezt már bemutattam az 1.2. részben, erre kiválóan alkalmas egy irányított, súlyozott teljes gráf. Ennek a gráfnak az élsúlyai többdimenziós vektorok, amelyeknek minden egyes komponense egy-egy (probléma-specifikus) tulajdonsgáként értelmezett súlyt tartalmaznak.

#### Az adatok előfeldolgozása, tisztítása

Ahhoz, hogy az irányított, súlyozott teljes gráfban értelmezni tudjuk az adott problémát, mindenkiéppen szükség van szakterület-specifikus tudásra, azaz a problémakör ismeretére. Ezt azért nem tudjuk automatizálni, mert sokszor a rendelkezésre álló adatok nem az egyes kapcsolatokhoz vannak rendelve, hanem például a csomópontokhoz. Ezeket a speciális eseteket le kell fordítani élsúlyozásra. Emellett természetesen lehetséges, hogy maga a problémakör olyan információkat is kezel, amelyekre nincs szükségünk. Ebben az esetben ezen adatokat ki kell szűrni, amihez szintén szakterület-specifikus tudásra van szükség.

A gráf csúcshalmazát viszonylag egyszerűen meg tudjuk határozni, hiszen általában a problémakörök esetében tudjuk, hogy milyen csomópontokat értelmezünk. Az élhalmaz esetében már nem ilyen egyértelmű a helyzet, de ha nem tudunk jó becslést adni, értelmezhetjük a teljes gráfot, mint bemenetet. Ez természetesen igen kölcsönös lehet mind tárkapacitásban, mind algoritmikus futási időben. Ezért lehet szükséges az élhalmaz csökkentése, azaz az útvonalválasztási szabályrendszer domainének szűkítése. A Pareto-elv<sup>1</sup> értelmében megpróbálhatjuk az élek egy részét kizártani a vizsgálatból, azaz törölni, vagy az algebra  $\phi$  elemével súlyozni. (A végtelenül súlyozáskor természetesen az él megmarad, de az adott algebra biztosan nem fogja használni, ami az optimalizáló algoritmus futását gyorsíthatja, valamint fontos megemlíteni, hogy élet csak olyan esetben tudunk törölni, ha egyik algebra sem használná.)

A bemenet értelmezése során tehát az igazán fontos feladat az élek súlyának és irányításának meghatározása. Mivel nagy hálózatok vizsgálásakor a gráf gyorsan kezelhetetlen méretűvé válhat, ezért mindenkor a vizsgálni kívánt algebráknak megfelelően érdemes a súlyozást és az irányítást elvégezni. Például ha egy algebra  $\preceq$  operátora egészeket *hatékonyabban*

---

<sup>1</sup>A Pareto-elv, más néven a 80/20 szabály kimondja, hogy számos jelenség esetén a következmények 80%-a az okok mindössze 20%-ára vezethető vissza.

tud összehasonlítani, mint valósokat, emellett a tényleges sílyok nagyságrendje miatt elhangolható a törtrész, akkor érdemes a ( pontos-, de akár lefelé kerekítéssel képzett) egész részekkel dolgozni.

### 3.1.2. Az optimalizálás függvénye

Miután generáltuk az irányított, vektor-sílyozott gráfot, szükség van egy függvényre, mely ki tudja választani utak egy halmazának legjobb útjait. Ezen legjobb utak algebránként változnak, ám az algoritmus lehet ugyanaz. Ilyen a következő algoritmus:

---

#### Algoritmus 1 Optimalizáló függvény

---

```

1: procedure GETPREFERREDPATH(Vertex u, Vertex v, Algebra a)
2:   paths  $\leftarrow$  GETALLPATHSBETWEEN(u, v)
3:   preferred  $\leftarrow$  [ ]
4:   for each path in paths do
5:     if  $a_W(path_i) \leq a_W(path)$  then
6:       preferred  $\leftarrow$  preferred + path
```

---

Természetesen ez a függvény nem implementálható ebben a formában, hiszen nincs részletezve, hogy hogyan működik a GETALLPATHSBETWEEN(*u*, *v*), de az általános felhasználhatósága látszódik az 5. sorban: az összes út közül a legjobbakat kiválasztani csupán az aktuális algebra alapján is képesek vagyunk. (Itt jól látszik egy algebra „jól viselkedése”, hiszen a GETALLPATHSBETWEEN() függvény akár a gráf méretében exponenciális méretű úthalmazzal is visszatérhet.)

A konkrét implementálás előtt gondoljuk meg, hogy érdemes az összes út előállítása, majd a minimálisak kiválasztása helyett csak a minimális utak megtalálását megpróbálni. Egy minimális út keresésére számtalan *mohó* megoldás ismert, melyek optimálisak és a gráf méretéhez képest polinom időben futnak. Nem triviális probléma azonban az összes ilyen út megtalálása. Erre a problémára használható egy módosított Dijkstra algoritmus.

### 3.1.3. A kimenet

A szimulációs folyamat kimenetén statisztikai elemzéseket szeretnénk lefolytatni, ezért ügyelni kell arra, hogy ez biztosítva legyen, illetve a folyamat lefutása közbeni értékes adatokat is gyűjteni, menteni és a kimeneten pedig feltüntetni szükséges.

### 3.2. A szimulátor

A szimulációs algoritmus a fent leírt irányított, súlyozott gráfot, mint bemeneten fut. Először meghatározzuk, hogy a konkrét probléma vizsgálata mekkora csúcshalmazt jelent, amelyből előáll egy  $G(V = n, E = n \cdot (n - 1))$  méretű gráf<sup>2</sup>. Ezután meghatározzuk azon éleket, melyet ezen a gráfot az útvonalválasztási szabályrendszer akármilyen két pont között legalább egyszer használt. Ezt természetesen meg tudjuk tenni, hiszen a kialakult utakat megfigyelni képesek vagyunk. Miután meghatároztuk a felhasznált élhalmazt, minden olyan élet, amelyet nem tartalmaz ez a halmaz kitörlünk a gráfból (Az élek törlésével nem eshet szét a gráf, hiszen a ponthalmazt megfigyeléssel kaptuk, így biztos, hogy összefüggő marad a gráf.). A megmaradó éleket súlyozzuk.

A szimuláció menete ezután a következő: minden pontpárra és minden vizsgált algebrára meghatározzuk (a módosított Dijkstra algoritmussal), hogy milyen ut(ak)at jelölnek ki. Az így meghatározott ut(ak)at feljegyezzük. Ezzel előállítunk két gráfot: az első az eredetileg vizsgált gráf, a másik a feljegyzett utakból felépülő. Ezután ezen két gráfot hasonlítjuk össze.

#### 3.2.1. A matematikai struktúrák implementálása

A különböző matematikai struktúrák felhasználása nem magától értetődő. Mivel bármilyen útvonalválasztási szabályt el lehet képezni, és viszonylag egyszerűen meg lehet alkotni az ehhez tartozó algebrát, nem minden egyszerű megfogalmazni a számítógép nyelvén ezeket. Ahhoz, hogy a módosított Dijkstra algoritmus futtatható legyen, négy alapvető funkcióval kell ellátni egy „szoftver-algebrát”: Egy súlyfüggvényel (`W(Route route)`), egy összegző függvényel (`bigOPlus(Weight w1, Weight w2)`), egy átjárhatatlanságot (végtelen súlyt) jelentő függvényel (`phi()`) és egy olyan függvényel, amely megadja, hogy szerinte mi a legjobb (legkisebb) súly (`best()`). Látszódik, hogy a matematikai struktúra nem ültethető át tisztán egy szoftverbé, nincs szükség például a súlyok alaphalmazára, ez nem az algebrához tartozik egy szimulátoron belül (persze ismerni kell a halmazt).

A másik absztrakt fogalom, amit a szimuláció során használok, az a gráf, csúcsaival és éleivel. Mivel itt a csúcsoknak már biztosan nem tulajdonítunk semmilyen információhorizontális szerepet (lásd a csúcsok adatainak érekre való átírását a 3.1.1. szakaszban), egy egyszerű (egész) azonosítóval jelöljük, míg az élek – összetettségük miatt – már bonyolultabb modellt igényelnek, melyek azonban mindenkor az adott problémától függenek és algebra-súly párokat tartalmaznak (adott algebra szerint mi a súlya az élnek).  
A gráfot ezután vagy éllistával, vagy szomszédossági mátrixszal jeleníthetjük meg. A szomszédossági mátrix a sűrű gráfoknál hatékonyabb, de az itt vizsgált problémák általában ritka gráfokkal modellezhetők<sup>3</sup>. Mégis a módosított Dijkstra algoritmust a mátrixos meg-

<sup>2</sup>Minden csúcsot minden csúccsal összekötünk és minden éről irányban irányítunk.

<sup>3</sup>Ha elfogadjuk, hogy *valamelyen* elv szerint optimális az útvonalválasztás, akkor nem lehet sok párhuzamos út  $u$  és  $v$  között, hiszen a redundancia nem hatékony az élszámok tekintetében.

adásra készítettem fel pusztán a programozástechnikai egyszerűsége miatt (természetesen a mátrixos megadásnak az optimalizált formáját használom, azaz csak a létező éleknél tárolok adatot).

### 3.3. A szimulációs eredmények kiértékelése

A szimulációs folyamat két gráfot eredményez, melyek összehasonításával szeretnénk meg-  
határozni, hogy az adott algebra mennyire képes leírni az eredeti (csupán megfigyelt) út-  
vonalmálasztási szabályrendszert.

#### 3.3.1. A vizsgalandó metrikák

A két gráfot – az eredetit és a generáltat – szeretnénk összehasonlítani, hogy mennyire hasonlítanak. A pontos egyezés eldöntése általában nehéz feladat, hiszen a gráfok izomorfiájának<sup>4</sup> eldöntése ugyan NP-beli (bővebben lásd az F.1.12. definíciót), de nem ismert hatékony módszer rá. Emellett nyilván az is szempont, hogy itt az is fontos és használható eredmény, ha nagyon hasonlítanak egymásra a gráfok (hálózatok), mert ez azt jelentené, hogy a nem ismert útvonalálasztási szabályt jól tudtuk becsülni az adott algebrával. Ez abban az esetben érdekes, ha az eredeti szabályrendszer bonyolult, vagy pl. egy mesterséges hálózat esetén nem automatizált: ekkor ugyanis a bonyolult szabályrendszert le lehet cserélni a szimuláció során használt policy primitív(ek)re. Így tehát még abban az esetben sem lenne érdemes izomorfiat vizsgálni, ha rendelkezésünkre állna egy erre alkalmas algoritmus.

Kétféle szempontrendszer szerint hasonlítom össze a gráfot: pontpáronként és globális hálózati mutatók alapján. Alapvető különbség van a két megközelítésben, hiszen különböző tulajdonságokat lehet megtudni a kétféle vizsgálatból.

A pontpárok közötti összehasonlítása során megvizsgálom, hogy a megfigyelt és a szimulált út mennyire hasonlít a lépésszám és az érintett köztes csomópontok alapján és ez alapján inkább jobb vagy inkább rosszabb a szimulált út, esetleg nem eldönthető (ez minden függ a problémától, ezért a kiértékelésnél is szükség van probléma-specifikus tudásra).

A globális hálózati metrikák közül a fokszámeloszlást fogom vizsgálni, valamint az élek számát, a hálózat átmérőjét és az élösszefüggőségét. A fokszámeloszlás és az átmérő összehasonlításra azért van szükség, mert ezek alapvető karakterisztikáját adják a hálózatnak. Ezzel feltárhatók olyan változások, melyek a pont-pont kapcsolatok vizsgálata során nem vettünk észre, például lehetséges, hogy a valóságban skálafüggetlen hálózatnak a fokszámeloszlása a szimuláció során már sokkal kiegyensúlyozottabb lesz. Hasonlóan fontos az élösszefüggőség, mellyel a gyakorlati használhatóságot mérhetjük, hiszen az eredetinél a szétesésre jóval érzékenyebb gráf esetén nem érdekes, hogy a többi szempont szerint „jól teljesít” az adott algebra.

---

<sup>4</sup>Két gráf izomorfijának bijektív struktúratartó leképezést értünk, lásd pontos definíciót az F.1.11 résznél.

Minden fent említett összehasonlítási szempontot pontozok a következő részekben meg-határozottak szerint és így minden szimulációs eredményt (algebrát) két pontszámmal fogok tudni mérni, egy globális- és egy pontpárokra vonatkozó mutatóval. Így könnyen meg tudom mondani, hogy az algebrák hogyan teljesítenek egymáshoz képest. A szempontoknál jellemzően háromféle pont adható: a legtöbb pont arra jár, ha szimulált eredmény a megfigyellett megegyezik, ennél kicsivel kevesebb pontot ér, ha az adott szempont szerint jobb, illetve a legkevesebb (akár negatív pont is lehetséges), ha rosszabb annál.

Ahhoz, hogy meg tudjuk mondani, hogy milyen pontszám az, ami már elég pontosnak számít, referencia szimulációkat futtatok. Egy ilyen referencia vizsgálat során tudom, hogy milyen útvonalválasztási szabályrendszer alakította ki az utakat a hálózatban, így tudom, hogy egy adott algebrának hogyan kéne teljesítenie. Ezután az ezen a gráfon futtatott szimulációs eredményeket ki tudom értékelni, így megkapom azt a pontértéktartományt, illetve küszöbértéket, ami mellett elfogadhatóan pontosnak tartok egy algebrát.

A következőkben leírt pontértékeket egy kezdeti pontozásból kiindulva a referencia szimulációk során finomhangolva határoztam meg. Emellett külön figyelmet fordítottam arra, hogy különböző mérőszámokat különböző súlyal szeretnék figyelembe venni.

### **Az algebra szerinti távolság pontozása**

Az algebra szerinti távolság pontértékét  $AL$ -lel jelölöm ( $AL$ : *AlgebraLength*). Az  $AL$  értékét úgy alakítom ki, hogy nagy hangsúlyt kapjon, hiszen ez az a tulajdonság, ami miatt az algebrákkal foglalkozom, amit le tudnak írni a policy-kból a matematika nyelvén. Ezért a pontosan becsült út 0 pontot, a rövidebb út azzal arányos pontot ér, amennyivel rövidebb (pl. a valós hálózat vizsgálatánál km-ben számított távolság nyereség). A megfigyeltnél hosszabb út is kialakulhat, hiszen az algebra lehet, hogy másik szempont szerint optimalizál, ez azzal arányos pontot ér, amennyivel hosszabb az út.

**3.3.1. Megjegyzés.** *Ha a megfigyelt hálózatban csak az összes útvonalat ismerjük és ebből egy élhalmazt kapunk meg, akkor természetesen nem triviális, hogy hogyan kezeljük az algebra szerinti távolságot, sőt a következő pontpárokra értelmezett pontokat sem. Nem tudjuk biztosan megmondani, hogy az élhalmazból melyik élek mentén jutunk el a pontpárok között. Abban az esetben viszont, ha a megfigyelés úgy történik, hogy az adott pontpárok közötti utakat ismerjük (figyeljük meg), akkor nincs ilyen probléma.*

### **A lépésszám szerinti távolság pontozása**

Hasonlóan az  $AL$  pontozáshoz, a lépésszám szerinti távolság pontértékét a  $HC$  (*HopCount*) jelöli. Ez a mutató jóval inkább mennyiségi, mint minőségi mutató, hiszen az algebra alapvető hozzáadott értékét (a speciális távolságmérést) figyelmen kívül hagyja, nem törődik a meghatározott út távolságával, csak nagyon áttételesen. Ezért a pontértékek rendre meg-egyezik a változással – esetleg a változásnak konstansszorosával.

**3.3.2. Megjegyzés.** *Értelemszerűen az  $AL$  és  $HC$  mutatók együttes értelmezése további információk kinyerését teszi lehetővé. Világos, hogy az algebra szerinti távolság fontos, de*

problémafüggő, hogy „milyen áron”. Hogyan értékeljük azt az eseményt, hogy az algebra szerinti távolság minimális csökkenése mellett a lépésszám egyről háromra nőtt? Problémától függően ez lehet, hogy nem jelent problémát, ám abban az esetben, ha ezt figyelembe kell venni, akkor az alábbi összefüggést érdemes vizsgálni:  $\frac{1}{(HC+1)} \cdot AL$ . A HC nem lehet negatív, hiszen a megfigyelt utak minden követlenek (azaz 1 a lépésszámuk), így a generált útnak 0 lépésből kéne állnia.

### A fokszámeloszlás pontozása

A fokszámeloszlás már globális mutató, melyet *DD*-vel (*Degree Distribution*) jelölök. Ez a tulajdonság olyan alapvető, hogy nem különböztetek meg „pontos”, jobb és rosszabb esetet, csak *hasonlót* és *különbözőt*, melyek pontértéke rendre 1 és  $\infty$ . A *DD* mutatót nem is összegezve számítom bele a végpontszámba, hanem szorzótényezőként, így az, hogy hasonlítsa a szimulált hálózat fokszámeloszlását a megfigyeltéhez egy szükséges feltétele annak, hogy értékelhető megoldást kapunk. Ezután a *DD* értéke a következő:

$$DD = \begin{cases} 1 & \text{ha megegyezik a fokszámeloszlás} \\ \infty & \text{különben} \end{cases}$$

### A hálózat-átmérő pontozása

A hálózat átmérője azért ad fontos információt, mert az átmérő a hálózat tömörségét mutatja, azaz azt, hogy maximálisan mennyi lépést kell megtenni a gráfban, ha tetszőleges két pont között akarunk útvonalat választani. Ha nagy az átmérő, az tipikusan azt jelenti, hogy könnyen lehet alternatív utakat találni, hiszen ezeket még nem találták meg, használták fel. Hasonlóan olyan hálózatokban, amelyek tömörek, az átmérőjük kicsit, nehéz „jó” alternatívákat keresni, hiszen – legalábbis a hop-szám tekintetében – csupa rövid út köti össze a pontpárokat. Emellett fontos megjegyezni, hogy az átmérő csak más hálózati mutatókkal együtt hordoz értelmezhető jelentést.

Ennek fényében a *GD* (*GraphDiameter*) jelölésű pontérték megegyezik a hálózat tényleges átmérőjével.

### Az élösszefüggőség pontozása

Egy gráf pont- illetve élösszefüggősége kiszámítható legfeljebb  $\binom{n}{2}$  maximális folyam keresésével. Nagamochi és Ibaraki megmutatta, hogy a probléma megoldására elegendő  $O(mn)$  idő is [17]. A pontösszefüggőségi számot  $\kappa(G)$ -vel, az élösszefüggőségi számot pedig  $\lambda(G)$ -vel jelölve felírhatjuk azt az egyszerű észrevételt, miszerint legalább annyi élet kéne elhagyni a gráf széteséséhez, mint pontot:  $\kappa(G) \leq \lambda(G)$  azaz egy  $k$  pontösszefüggőségből következik a legalább  $k$  élösszefüggőség. Emellett fontos megjegyezni, hogy a szimulációs eredmények kiértékelése folyamán a pontösszefüggőséget nem fogom külön vizsgálni, mert a legtöbb probléma vagy úgy áll elő, vagy olyan alakra transzformálható, ahol nem lehet értelmezni egy csomópont törlését. Természetesen azon speciális esetekben, ahol ez nem így van és tényleges jelentést is hordoz a  $\kappa(G)$ , ott ezt a mutatót is érdemes vizsgálni.

Mivel az élösszefüggőséget könnyen el lehet torzítani, nem tisztán a  $\lambda(G)$ -t határozom meg. A valóságban megfigyelhető hálózatok túlnyomó többsége skálafüggetlen, és mint olyan, sok alacsony fokszámú csúcst tartalmaz, jellemzőn elsőfokúakat is, amikor  $\lambda(G) = 1$ , de a gráf „mag”-járól, annak tömörségéről nem ad információt. Ezért a  $C$  (EdgeConnectivity) pontérték a csomópontok fokszámeloszlás szerinti felső 20%-ának minimális fokszáma lesz. Ez megegyezik ezen 20% élösszefüggőségével, hiszen az a csomópont, aminek pontosan ennyi a fokszáma, egy ekkora vágással eltávolítható a gráfból.

### Az élszám pontozása

Az élszám pontozását  $EC$ -vel (EdgeCount) jelölöm. A  $EC$  olyan a  $DD$ -nek, mint a  $HC$  az  $AL$ -nek, azaz a hasonló típusú mutatók közül ez inkább a mennyiségi, a  $DD$  a minőségi mutató. Éppen ezért kevesebb értékkel, jelentéssel bír az  $EC$ . Ha pontosan ugyanannyi élet használ a szimulált gráf, mint a megfigyelt, akkor  $EC = an + b$  (a csúcsszámmal lineáris a kapcsolat, jellemzően  $n/20$ ), ha nem, akkor  $EC = f(C)$ , azaz függ az összefüggőségtől, méghozzá úgy, hogy

$$0 > \frac{\partial EC}{\partial C}$$

Ez azt jelenti, hogy az élösszefüggőség pontértékének növekedésével csökken az élszám pontértéke. Erre azért van szükség, mert az élösszefüggőség magas értékéből következik a magas élszám, és ugyanazt a jelenséget (sok él van a gráfban) nem szeretnénk duplán értékelni.

### A globális mutatók egymásra hatása, együttes értékük

Az összes globális mutatóról elmondható, hogy tényleges értéket, értelmezhető jelentést csak akkor hordoznak, ha egymással kapcsolatban vannak értelmezve, mégpedig a megfelelő hatásokat és ellenhatásokat figyelembevéve. A  $GV$  (GlobalValue) értéket a következő összefüggések<sup>5</sup> befolyásolják:

$$\begin{aligned} 0 &< \frac{\partial GV}{\partial GD} \\ 0 &< \frac{\partial GV}{\partial C} \\ 0 &> \frac{\partial GV}{\partial EC} \end{aligned}$$

Egy ilyen függvény a következő:

$$GV = \frac{DD \cdot GD \cdot C}{ln(EC)}$$

---

<sup>5</sup>A pozitív derivált azt jelenti, hogy a több jobb, és fordítva.

### **3.4. Valós hálózatok vizsgálata a keretrendszer segítségével**

Mivel a diplomamunka feladatkiírásában nem volt benne, hogy saját szimuláltort kell használnom, az ezzel kapcsolatos technikai részletek a Függelékben (az F.4. rész) található.

A 2. fejezetben definiált algebrák modellezése után valós hálózatokon vizsgálhatjuk a viselkedésüket, hogy hogyan alakítanak ki utakat. Ezután tetszőleges algebrák lexikografikus szorzatát is szimulálhatjuk, így még pontosabb és változatosabb modellt kapunk.

### **3.5. Összefoglaló**

Ebben a fejezetben leírtam egy olyan keretrendszer, melynek segítségével szimulálás segítségével össze lehet hasonlítani különböző algebrákat ugyanazon a problémán. Specifikáltam ezt a rendszert, felvázoltam az elvárt célokat. Ezután leírtam az általános problémák előfeldolgozását, a tisztított és előkészített adatok alapján a szimuláció pontos menetét. Meghatároztam a szempontokat, ami alapján a szimulációs eredményeket értékelni lehet, ehhez pedig kidolgoztam egy pontrendszer, mely figyelembe veszi, hogy az elsődleges cél a megfigyelt útvonalválasztási szabályok visszafejtése, így a pontozásban a helyesen megtalált utak érik a legtöbbet, utána következnek az eredetinél jobb utak és legutoljára az eredeti rosszabb utak.

A pontrendszer kidolgozását referencia szimulációk futtatásával támogattam, melynek segítségével már nem csak az algebrákat tudom összehasonlítani egymással, hanem egy abszolút mércét is kaptam.

## 4. fejezet

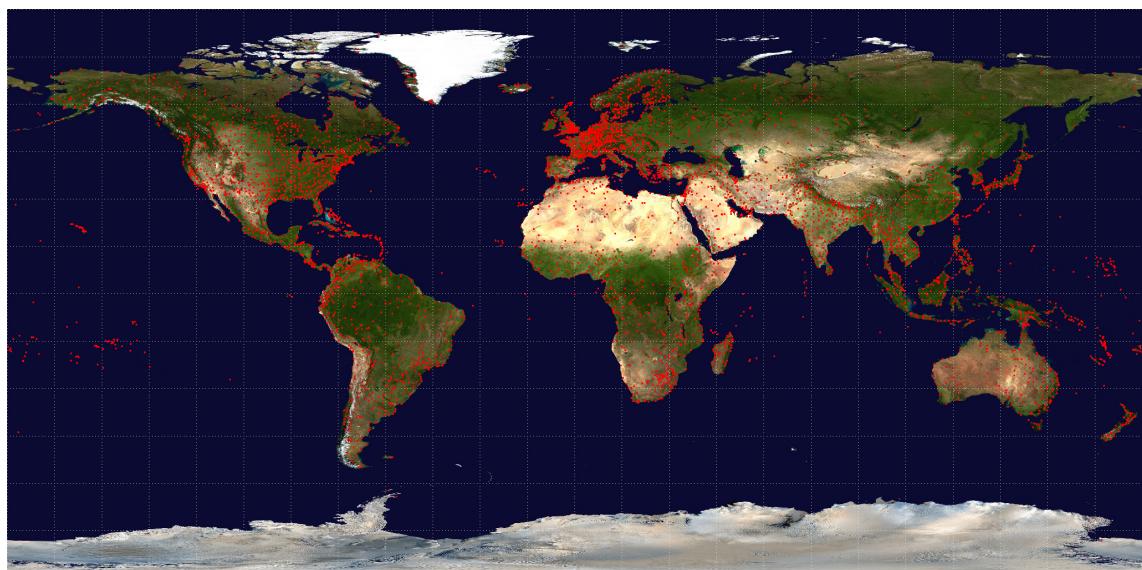
# Repülési hálózat vizsgálata

Ebben a fejezetben az openflights.org oldalon elérhető repülési adatbázis feldolgozását és ezen adatokon futtatott szimulációt mutatom be.

### 4.1. Az adatok

Az openflights lehetőséget biztosít különféle repülési adatok interaktív keresésére, regisztráció után pedig akár statisztikákat készíthetünk saját utazásainkról. Emellett elérhető a mindenkor aktuális, teljes adatbázisa is, amit többek között hivatalos forrásokból, illetve a felhasználóktól gyűjt. Az oldalon elérhető adatbázis világszerte mindenhonnan gyűjt és tárol reptéri, repülőtársasági és repülési út adatokat. Több, mint 9000 reptér, 19000 társaság és közel 60000 repülési út érhető el. Részletes adatok a 4.1.1, a 4.1.3, és a 4.1.2 táblázatokban.

#### 4.1.1. Repterek



**4.1. ábra.** Az OpenFlights adatbázisában szereplő repterek.

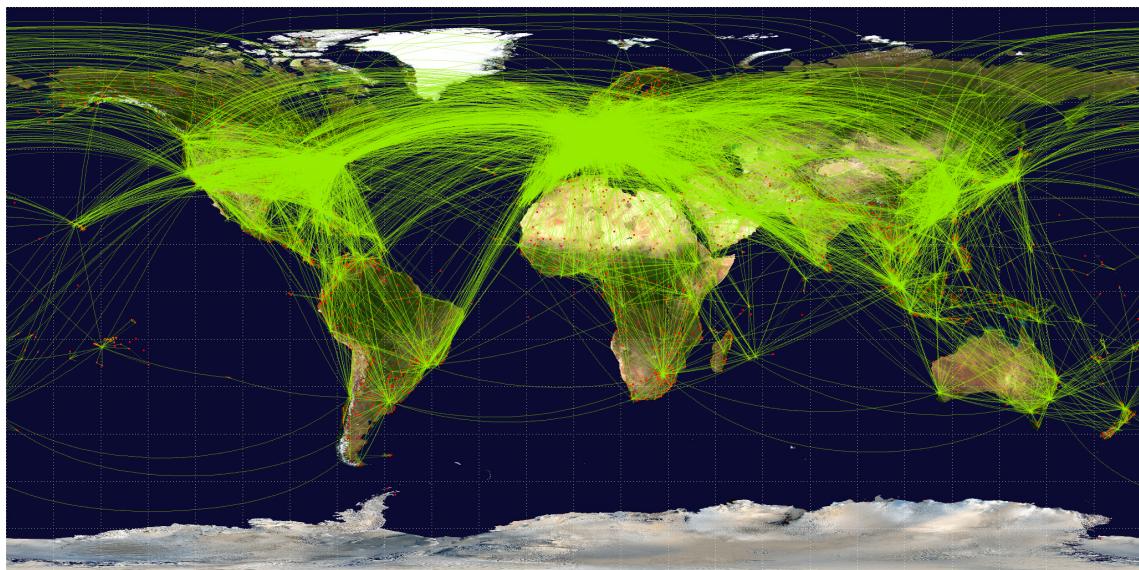
A rendszer tárolja a repterek *nevét*; a *várost*, ahol található; a *szélességi-* és *hosszúsági*

fokokat, illetve a *tengerszint feletti magasságot*. Emellett elérhető a repülési hatóságok által használt 3 karakter hosszú *IATA*<sup>1</sup>/*FAA*<sup>2</sup> azonosító is és az OpenFlights által használt decimális azonosító is. A legutolsó, 2014 januári frissítéskor az OpenFlights adatbázisa 9167 repteret tartalmazott világszerte, ezt mutatja a 4.1 ábra.

Attribútum	Leírás
Airport ID	Egyedi OpenFlights azonosító.
Name	A reptér neve.
City	Az a város, amit a reptér „kiszolgál”.
Country	Az ország vagy terület neve, ahol reptér található.
IATA/FAA	3 betűs FAA kód az USA-beli reptereknek vagy 3 betűs IATA kód, minden más esetben.
ICAO	4 betűs ICAO <sup>3</sup> kód.
Latitude	A reptér szélességi foka: decimális szám (fokban mérve), általában 6 szignifikáns jegyig. A Déli féltekén negatív, az Északon pozitív.
Longitude	A reptér hosszúsági foka: decimális szám (fokban mérve), általában 6 szignifikáns jegyig. A Nyugati féltekén negatív, a Keleten pozitív.
Altitude	A tengerszint feletti magasság, lábban mérve (1 láb ~ 0,30 méter).

**4.1.1. táblázat.** Az OpenFlights adatbázisában elérhető reptéri adatok.

#### 4.1.2. Repülési útvonalak



**4.2. ábra.** Az OpenFlights adatbázisában szereplő repülési útvonalak.

A repülési útvonalak leírása öt attribútumot tartalmaz: az *üzemeltető repülőtársaság* (név és OpenFlights azonosító); a *forrás reptér*; a *cél reptér*. Emellett feljegyzik a *megállások számát*<sup>4</sup>, illetve az ebben a viszonylatban használt *repülőgép típusokat*. Ha két reptér között oda-vissza is van járat, akkor ez az adathalmazban két bejegyzésként jelenik meg. Jelenleg az OpenFlights adatbázisa 59637 útvonalat tartalmazott 3285 reptér és 531 társaság<sup>5</sup> között világszerte, ahogy a 4.2 ábra is mutatja.

<sup>1</sup>International Air Transport Association: Nemzetközi Légi Szállítási Szövetség

<sup>2</sup>Federal Aviation Administration: Szövetségi Légügyi Hivatal (USA)

<sup>4</sup>Az összesen 59637 bejegyzésből 6 esetében van 1 megállás és 1 esetben 2 megállás, ezért ezt az adatot

Attribútum	Leírás
Airline	Az üzemeltető társaság 2 betűs (IATA) vagy 3 betűs (ICAO) kódja.
Airline ID	Az üzemeltető társaság egyedi OpenFlights azonosítója.
Source airport	A forrás reptér 3 betűs (IATA) vagy 4 betűs (ICAO) kódja.
Source airport ID	A forrás reptér egyedi OpenFlights azonosítója.
Destination airport	A cél reptér 3 betűs (IATA) vagy 4 betűs (ICAO) kódja.
Destination airport ID	A cél reptér egyedi OpenFlights azonosítója.
Codeshare	Igaz, ha a járat „codeshare”, azaz nem utasszállító járat, különben üres.
Stops	A megállások szám (nem az átszállásokat jelenti, a „0” jelenti a direkt járatot).
Equipment	A viszonylatban használt repülőgép típusok 3 betűs kódjai.

**4.1.2. táblázat.** Az OpenFlights adatbázisában elérhető adatok a repülési útvonalakról.

#### 4.1.3. Repülőtársaságok

A repülőtársaságokról tárolt adatok többek között tartalmazzák a hivatalos *IATA azonosítót*; a cég *nevét*; az *országot*, ahol be van jegyezve; illetve, hogy *aktív-e* még a társaság. Emellett természetesen itt is definiáltak egy saját (OpenFlights) *azonosítót*.

Attribútum	Leírás
Airport ID	Egyedi OpenFlights azonosító.
Name	A társaság neve.
Alias	A társaság egyéb megnevezése.
IATA	2 betűs IATA kód.
ICAO	3 betűs ICAO kód.
Callsign	A társaság hívójele.
Country	Az ország vagy terület neve, ahol a társaság be van jegyezve.
Active	Igaz, ha a társaság jelenleg is, vagy nemrég még működött (nem megbízható).

**4.1.3. táblázat.** Az OpenFlights adatbázisában elérhető adatok a repülőtársaságokról.

### 4.2. A szimuláció

A 3 fejezetben ismertetett keretrendszer használva fogom megvizsgálni a fent bemutatott adathalmazon a  $\mathcal{S}$  (Shortest algebra), a  $\mathcal{L}$  (LeastHop algebra:  $\mathcal{L} = (\mathbb{N}, \infty, +, \leq)$ , legkevesebb csomópontot érintő út), az  $\mathcal{O}$  (Összekötő-keresés: 2.2.1. rész) és a  $\mathcal{K}$  (Korai-elfogadó-keresés: 2.2.1. rész) algebrákat. Mivel az utolsó kettő algebra nem jól viselkedő<sup>6</sup> (lásd az 1.3.1. részt.), ezért csak a  $\mathcal{S}$ ,  $\mathcal{L}$ ,  $\mathcal{OS}$ ,  $\mathcal{KS}$ ,  $\mathcal{OL}$  és a  $\mathcal{KL}$  algebrákat fogom ténylegen szimulálni, mert a lexikografikus szorzatban már csak egy adott útvonalhalmazból kell kiválasztani a szorzat első tényezője szerinti legjobbat.

#### 4.2.1. Az adatok előfeldolgozása

A modellezendő hálózat első megközelítésben a minden csomópontból alkotott  $K_{9167}$  teljes gráf. Ez természetesen feleslegesen nagy hálózatot jelent ( $(^{9167})_2 = 42\ 012\ 361$  élű gráf.),

a szimuláció során semmilyen módon nem használom fel.

<sup>5</sup>Ugyan a légitársaság és reptér adatbázisokban ennél jelentősen több bejegyzés van, ám azok között vannak nem aktív társaságok, illetve bezárt repterek.

<sup>6</sup>Világos, hogy  $\geq$  operátorú algebrák maximumot keresnek, ami ezeknél a konkrét algebráknál maximális utak keresését jelenti. Viszont az F.1.2. téTEL értelmében csak akkor tudunk maximális utakat keresni, ha a gráf DAG, ami nyilvánvalóan ebben az esetben nem áll fenn, hiszen ez azt jelentné, hogy egy reptérről indul soha nem juthatunk vissza ugyanoda.

emellett az olyan, leíró jellegű jellemzők, mint a valódi hálózat átmérője, vagy az élőssze-függőség elvesznének. Lévén a repülési hálózat skálafüggetlen – sok olyan csomópont van, ahova csak adott, kis számú, közeli csomópontokból indulnak járatok, így nem érdemes összekötni távoli csomópontokkal. Ezenkívül a repülésnél vannak olyan szempontok, amelyeket az útvonalválasztásnál figyelembe vesznek, ám én ezek pontos információk hiányában nem tudom kezelni – pl. csak adott hosszúságú utat tehet meg egy adott típusú repülőgép az üzemanyagtartálya méretétől függően.

A fentiek miatt a szimulált hálózatot a következőképpen határozom meg: pontjai a repterek, élei pedig, az adatbázisban szereplő össze, megállás nélküli repülőút. Így magától értetődően csak olyan élek lesznek a szimulációban, amelyek „átrepülhetők”, hiszen volt már, hogy átrepülték. Emellett pedig ez a hálózat hűen tükrözi a valóságot, hiszen feltehetjük, hogy csak azon élek nincsenek behúzva, amelyeket repüléstechnikai okokból nem is lehet, különben valaki, valamikor már repült volna azon az úton.

### **Az élsúlyok meghatározása**

Ahogy a 3.1.1. részben kifejtettem, a vizsgálni kívánt algebráktól függően határozom meg az élsúlyozást, valamint a súlyozás során az eredményt nem jelentősen befolyásoló és a futást gyorsító egyszerűsítésekkel élıhetek.

A  $\mathcal{S}$  algebra miatt szükséges a csomópontok valós távolságára. Ezt a repterek szélességi és hosszúsági koordinátái alapján a Haversine formulával számolom. Annak érdekében, hogy pontos, de mégsem túl nagy számokat kapjak a távolságot [km]-ben mért egész számmal jelölöm (az akár [m]-ben mért egytizedes pontosság helyett).

A  $\mathcal{L}$  algebra pusztán az élek létét használja fel, így minden él súlya 1.

Az  $\mathcal{O}$  algebra a népszerű reptereket részesíti előnyben, ami azt jelenti, hogy itt egy csúcs érték → élsúly konverzióra van szükség. Mivel a gráf irányított, nem kell azzal törődni, hogy egy esetleges hibás megadás esetén az él súlyát akkor is terheli a népszerű reptér, amikor onnan elfelé haladunk. Egyszerűen minden csomópontra, a befelé mutató éleket a csomópont népszerűsége szerint súlyozunk. Ez a tényleges megvalósítást tekintve azt jelenti, hogy a repterekre érkező és az onnan induló járatok, tehát a reptér teljes forgalma adja meg a csomópont népszerűségét.

A  $\mathcal{K}$  algebra esetén azon az úton fogunk haladni, amin a legtöbben haladtak, így minden élnek a súlyát úgy határozzuk meg, hogy a megfigyelések szerint hányszor repültek az adott viszonylaton.

### **4.2.2. A szimuláció menete**

A repülési adatok ismeretében nem pontosan a 3.2. részben ismertetett módon szimulálok. A gráf élhalmaza a repülési adatokból kapott utak, és ezen a gráfon a 10 legjelentősebb légi-társaság útvonalait vizsgálom meg. Bár az előző fejezetben ismertetett szimulátor minden pontpárra előírná az útvonal meghatározását, de jelen esetben a skálafüggetlenség miatt a

szimulált utak túlnyomó többsége – a fokszámeloszlás szerinti „farkok” csúcsai kis fokszámúak – nem is lehet más, mint a megfigyelt út. Ez eltorzítaná a szimulációt, hiszen magas pontot ér, ha pontos a becslés. Ezzel szemben, ha csak a 10 legtöbb járatot üzemeltető társaságot veszem figyelembe, akkor a 3.3.1. részben leírt *maghoz* hasonlóan a lényeget kiemelve tudom vizsgálni a problémát: a 10 legtöbb járatot indító társaság bonyolítja le a repülőutak 24%-át.

### 4.3. Az eredmények

A szimulációs eredmények kiértékelését közönséges táblázatkezelő szoftver segítségével végeztem. minden algebra által generált gráfról kétféle adathalmazt dolgoztam fel: egy „útlistás” és egy csúcshalmazos reprezentációt.

Az útlistás megadást a generált utak alapján állítottam elő, négy adatot tartalmaz: a *forrás*- és *cél* csomópontokat, valamint a lépésszámot és az algebra szerinti távolságot – jelen esetben a repterek fizikai távolságát. Ezekből az adatokból közvetlenül számíthatók az *AL* és a *HC* mutatók.

A csúcshalmazos reprezentáció minden felhasznált csúcsra tartalmazza a csomópont ki- és befokát, valamint ezen mutatók átlagát. Azért volt célszerű előállítani ezt az adathalmazt is, mert innen egyszerűen meghatározható a csomópontok ki- és befoka alapján a kétdimenziós fokszámeloszlás, illetve az egydimenziós átlagfokszám-eloszlás. Ebből a forrásból egyszerűen számíthatók a *GD*, a *C* és az *EC* mutatók, amiből pedig a *GV*.

Ebben a részben elsőként a referenciaiként szolgáló szimulációt mutatom be, majd bemutatom minden algebra szimulációs eredményét, a fejezet végén pedig egy táblázatban foglalom össze az eredményeket.

#### 4.3.1. A Shortest algebra

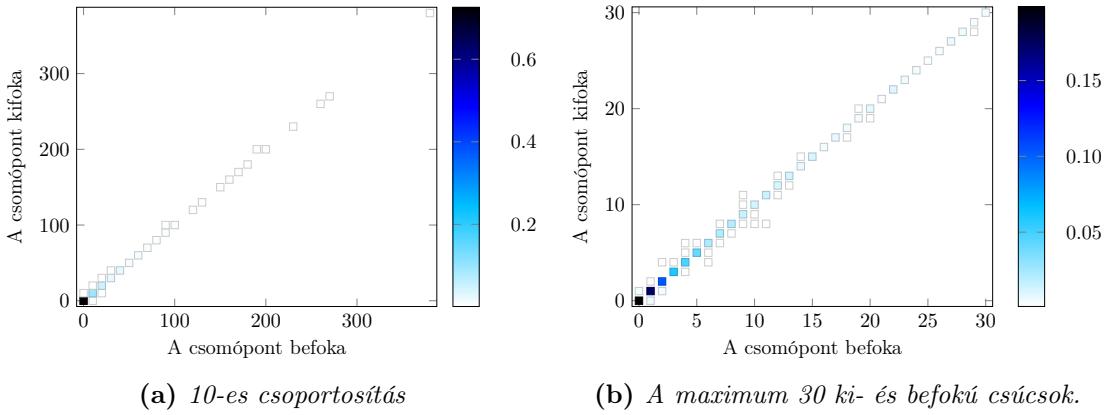
A legrövidebb utakat előnyben részesítő policy algebrája áll várhatóan a legközelebb a valós hálózathoz, hiszen várhatóan a fizikai távolság az útvonalválasztást leginkább befolyásoló tényező.

A  $\mathcal{S}$  algebra által generált gráfban 152 pontpár között sikerült javítani a távolságon. Az  $AL = 152$  és  $HC = 182$  így a 3.3.2 képlet szerint

$$AL-HC = 48\frac{1}{6}.$$

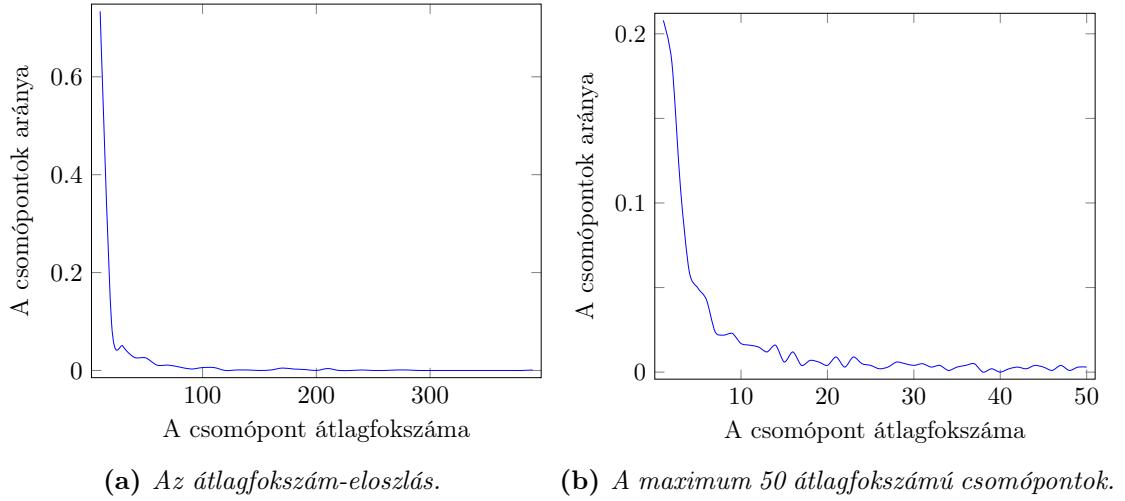
A globális mutatók közül a kétdimenziós fokszámeloszlás a 4.3. ábrán látható. A 4.3. (a) diagramon tízesével csoporthoztattam a fokszámokat, azaz az origóhoz legközelebbi jelölés

a 10, vagy annál kevesebb fokszámú csúcsokat jelenti. A 4.3. (b) ábrán egy „kinagyított” kép látható, csoportosítás nélkül.



**4.3. ábra.** A  $\mathcal{S}$  algebra által generált gráf fokszámeloszlása.

Látható, hogy a csomópontok ki- és befoka minden pontra közel megegyezik. Az átlagfokszámeloszlás a 4.4. ábrán látható. Hasonlóan a másik diagramhoz, itt is egy csoportosított ((a) diagram) és egy kinagyított ((b) diagram) látható. Az ábráról leolvasható, hogy a generált gráf skálafüggetlen, így  $DD = 1$ .



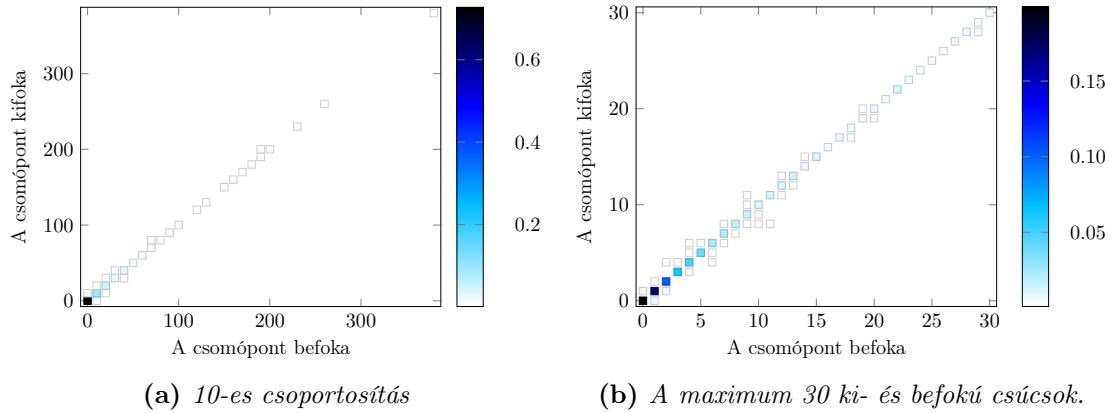
**4.4. ábra.** A  $\mathcal{S}$  algebra által generált gráf csomóponthonkénti átlagfokszámeloszlása.

A  $GD$  (átmérő) mutató 3, mert a legmagasabb lépésszámú út 3 lépést tartalmaz. Az előszefüggőségi mutatót a felső 20%-ból számítva  $C = 16$ . Az  $EC$  (élszám) pontértéke 9,5872 ( $\ln(14578)$ ). Így a Shortest algebrára a

$$GV = \frac{DD \cdot GD \cdot C}{\ln(EC)} = \frac{1 \cdot 3 \cdot 16}{9,5872} \approx 5,006.$$

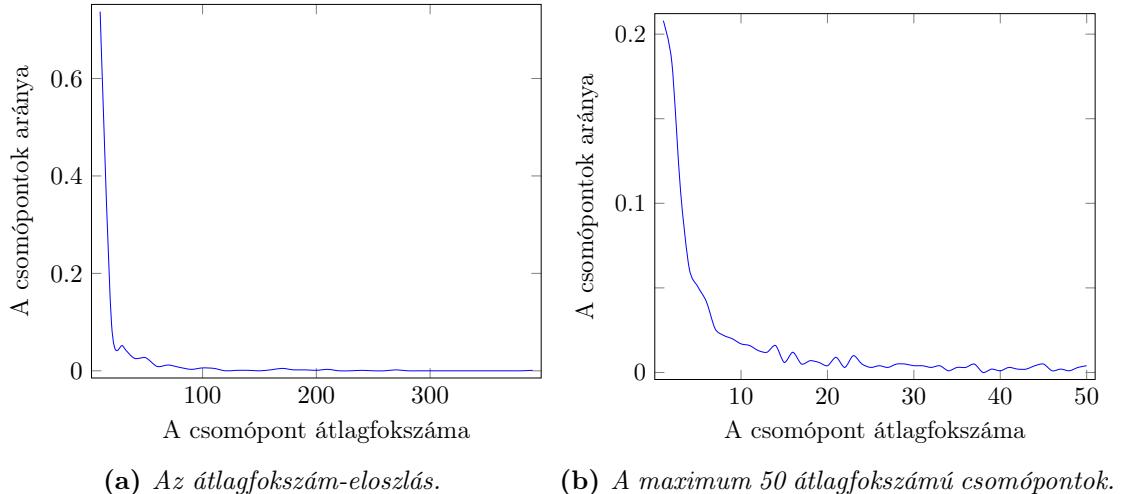
### 4.3.2. A LeastHop algebra

Abban az esetben, ha a lépésszám szerint optimalizálunk, a fizikai távolságokon nem tudunk javítani, hiszen a megfigyelt utak közvetlen járatok, azaz egy lépésben elérik a céljukat, ennél pedig nem tudunk jobbat találni. Így  $AL = HC = 0$ . Ez természetesen várható volt, nem is a fizikai távolságon javítás a célja a  $\mathcal{L}$  algebrának. A 4.5. ábrán látszódik a kétdimenziós fokszámeloszlás. Ennél az algebránál a csomópontok több, mint 73%-a legfeljebb 10 fokú. Nyilván  $DD = 1$ .



**4.5. ábra.** A  $\mathcal{L}$  algebra által generált gráf fokszámeloszlása.

Az erősebb skálafüggetlenségből következik, hogy a hálózat „magja” tömörebb, jobban össze vannak fűződve a csomópontok, hiszen ahol csak lehet, közvetlen utat jelöl ki az algebra.



**4.6. ábra.** A  $\mathcal{L}$  algebra által generált gráf csomóponthonkénti átlagfokszám-eloszlása.

A  $\mathcal{L}$  algebra által generált gráf globális adatai:  $GD = 1$ , hiszen minden út 1 lépésből áll. A  $C = 16$ .  $EC = \ln(14396) \approx 9,574$  és így

$$GV \approx 1,671.$$

### 4.3.3. A ShortestLeastHop és a LeastHopShortest algebrák

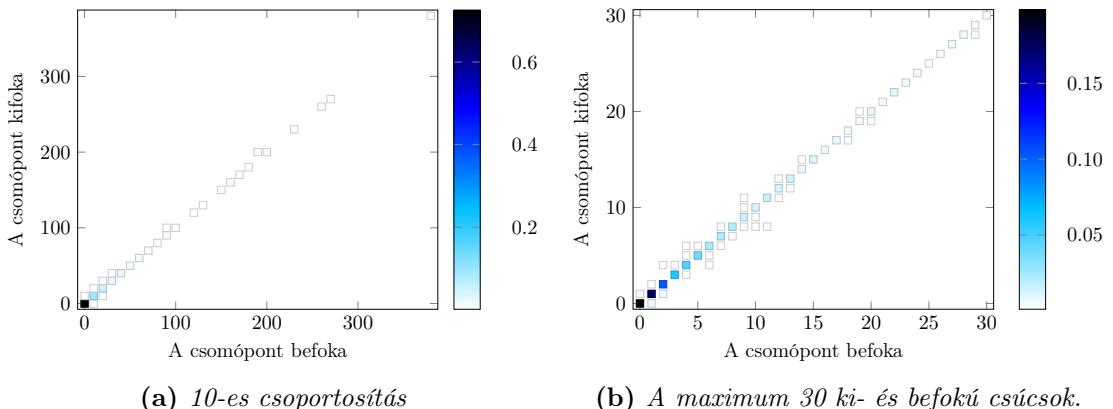
Bár szimuláltam a ShortestLeastHop- és a LeastHopShortest algebrákat is, ám ezek eredménye rendre megegyeznek a LeastHop és a Shortest algebrák eredményeivel, ezért itt nem tüntetem fel. Ez azért lehetséges, mert pl. a Shortest algebra szimulációjával nem feltétlenül egyértelmű a generált gráf. Ha van több legrövidebb út is, akkor véletlenszerűen választunk a legrövidebb utak közül, hiszen a fontos szempont az út hossza, a lépésszám nem számít ebben az esetben. Azt az eljárást, hogy legrövidebb utak közül választunk egyet véletlenszerűen, le lehet egyszerűsíteni úgy, hogy az összes út közül az „első” választjuk (ezt szoftveresen úgyis valamilyen listában vagy tömbben tároljuk), ezzel megspórolhatjuk a véletlenszám-generálást. Így viszont előfordulhat, hogy a tárolt legrövidebb utak közül az „első” éppen a másik algebra szerinti legjobb.

### 4.3.4. Az Összekötő-keresés- és Korai-elfogadó-keresés-LeastHop algebrák

Ezek az algebrák teljesen megegyeznek a LeastHop algebrával, ugyanis egy csomópontpár között 1 lépésben csak a közvetlen él van, amit a LeastHop algebra megtalál, így soha nem lesz szerepe az útvonalválasztásban sem az  $\mathcal{O}$ , sem a  $\mathcal{K}$  algebráknak.

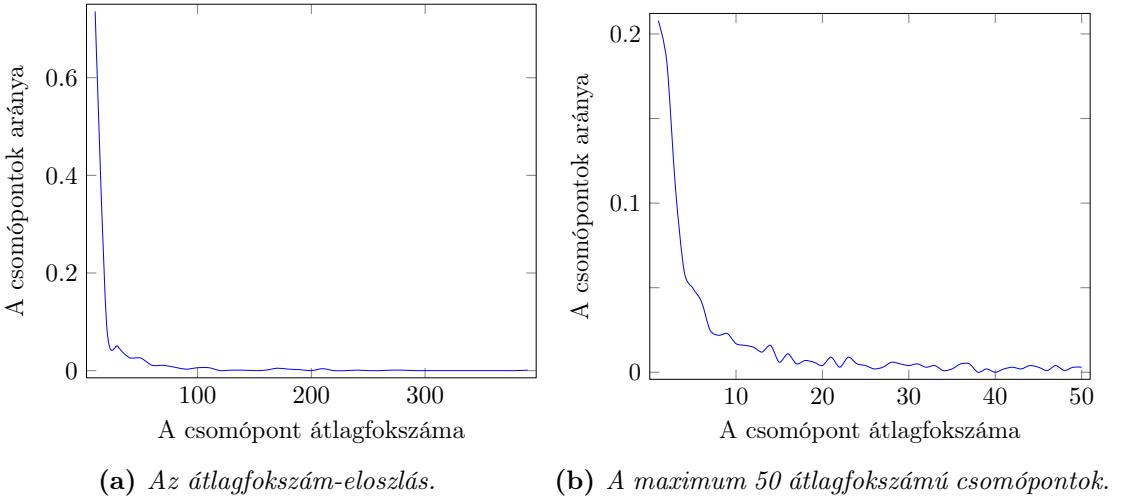
### 4.3.5. Az Összekötő-keresés-Shortest algebra

Az Összekötő-keresés-Shortest algebra egy pontpár közötti legrövidebb utak közül a legforgalmasabb repterek felé haladó utat fogja választani. Ezzel két alapvető tulajdonságot határozza meg, azaz a gráf „magja” és az élszám nagy lesz, hiszen két alternatív út közül azt választja, amelyik a több repteret érinti.



**4.7. ábra.** Az  $\mathcal{OS}$  algebra által generált gráf fokszámeloszlása.

A 4.7. és a 4.8 diagramokról leolvasható, hogy a generált gráf skálafüggetlen:  $DD = 1$ . Az  $AL = 152$  és  $HC = 185$ , így ezek a mutatók szinte megegyeznek a Shortest algebrá-



**4.8. ábra.** Az  $\mathcal{OS}$  algebra által generált gráf csomópontonkénti átlagfokszám-eloszlása.

éval, így

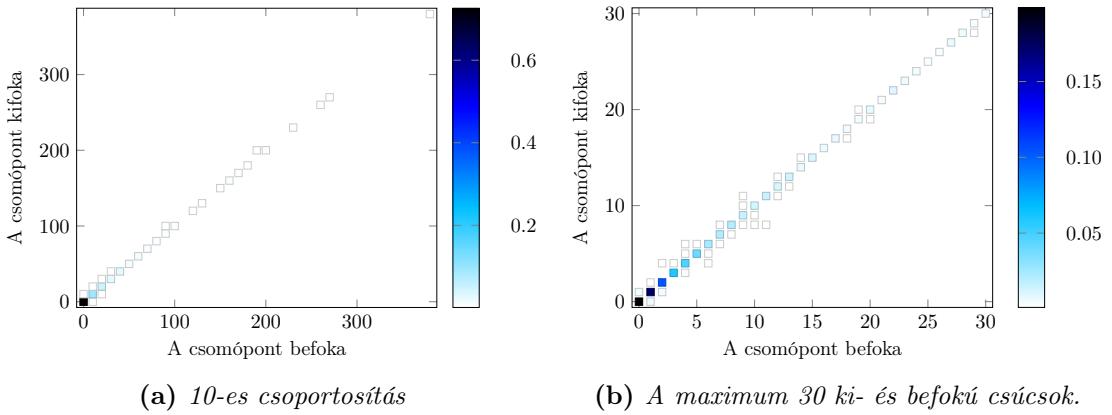
$$\text{AL-HC} = 47 \frac{11}{12}.$$

A  $GV$  globális mutató viszont romlott a  $\mathcal{S}$  algebráéhoz képest:

$$GV \approx 5,01.$$

A  $C = 16$  és az  $EC = \ln(14396) \approx 9,57$  értékek megegyeznek, viszont a  $GD = 3$ . Az eloszlás szinte teljesen megegyezik a Shortest algebráéval, hiszen „azon alapszik”, viszont ez az algebra az összefüggőséget növeli, hiszen minél több csomópontot szeretne érinteni egy adott hosszú úton, így nő a gráf átmérője is. A skálafüggetlenség miatt a „mag” annyira tömör, hogy nem látható az algebra összefüggőségnövelő hatása ( $C = 16$  nem változott).

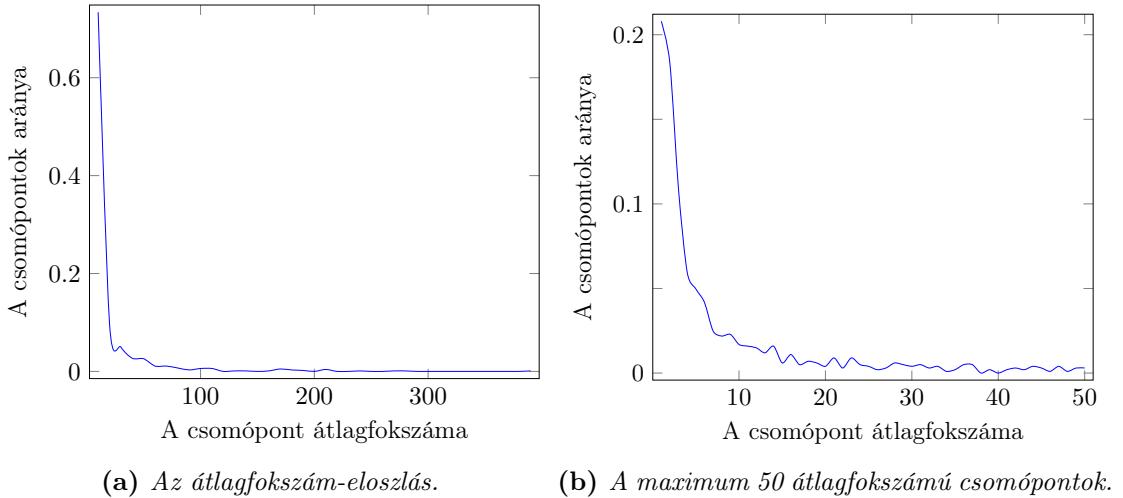
#### 4.3.6. A Korai-elfogadó-keresés-Shortest algebra



**4.9. ábra.** A  $\mathcal{KS}$  algebra által generált gráf fokszámeloszlása.

Ez az algebra a legrövidebb utakból azokat használja, amit előtte – mások – már sokan használtak. Ezt természetesen összefüggésben van az Összekötő-keresés algebrával ám itt

nem számít a reptér tényleges népszerűsége.



**4.10. ábra.** A  $\mathcal{KS}$  algebra által generált gráf csomóponthonkénti átlagfokszám-eloszlása.

A 4.9. ábrán látható a  $\mathcal{KS}$  algebra által generált gráf fokszámeloszlása, a 4.10. ábrán pedig a gráf átlagfokszám-eloszlása. A diagramokról leolvasható, hogy  $DD = 1$ .

A  $\mathcal{KS}$  algebra mutatói természetesen nagyon hasonlítanak a  $\mathcal{S}$  algebráéra, hiszen minden esetben a Shortest algebra adja az utakat, amiből választani kell egy legjobbat. Az  $AL = 152$  és  $HC = 185$  így a 3.3.2 képlet szerint

$$AL-HC = 47, 95.$$

A gráf átmérője 4, mert van egy 4 lépéses út a gráfban:  $GD = 4$ . A  $C = 4$  megegyezik, ám az  $EC = \ln(14581) \approx 9,587$ , ami valamivel több, mint az egyszerű Shortest algebra esetén, így

$$GV \approx 6,67.$$

#### 4.3.7. Az algebrák értékelése

Algebra	$AL$	$HC$	$AL - HC$	$DD$	$GD$	$C$	$EG$	$GV$
Referencia	0	0	0	1	1	16	9,574705669	1,671069645
$\mathcal{S}$	152	182	48,16	1	3	16	9,587268822	5,006639627
$\mathcal{L}$	0	0	0	1	1	16	9,574705669	1,671069645
$\mathcal{OS}$	152	185	47,916	1	4	16	9,574705669	5,013208934
$\mathcal{KS}$	152	185	47,95	1	4	16	9,58747459	6,675376231

4.3.1. táblázat. A szimulációs eredmények

A 4.3.1. táblázatban foglaltam össze a szimulációs eredményeket. Ahogy a 3.3.1. részben kifejtettem, egy algebra által generált hálózat pontosságát két tényező határozza meg. Az egyik a  $GV$ , ami globális mutatókból tevődik össze és minél kisebb az abszolútértéke, annál jobban közelíti meg az eredeti problémát. A másik mutató az  $AL - HC$ , ami a pontpárokhoz kapcsolódó mutatókat kapcsolja össze és minél nagyobb, annál többet lehet javítani az eredeti hálózathoz képest. Természetesen ezen két mutatót egyszerre kell figyelembe venni, pontosabban a legjobb  $GV$  értékű algebrák közül a legmagasabb  $AL - HC$  értékű algebra a legjobb.

Az eredményekből arra lehet következtetni, hogy bár a lépésszám szerinti elgrövidebb utakat választó  $\mathcal{L}$  LeastHop algebra közelíti meg a leginkább az eredeti hálózatot, ezzel az algebrával nem tudunk javítani. Ezzel szemben a fizikai távolság szerint optimalizáló  $\mathcal{S}$  Shortest algebra szintén elég pontos közelítése az eredeti hálózatnak, de közel 150 úton lehet még javítani. Emellett meg kell jegyezni, hogy az is kiolvasható a táblázatból, hogy az  $\mathcal{OS}$  algebra szinte megegyezik a  $\mathcal{S}$  algebra eredményével, ugyanakkor bonyolultabb. Mivel mindenkorral elég pontos becslés, feltehető, hogy a repülőtársaságok egy  $\mathcal{OS}$  algebrához hasonló szabályrendszer szerint választják meg az útvonalakat. Ebben az esetben érdemes lenne nekik áttérni az egyszerű  $\mathcal{S}$  algebrára, hiszen a két hálózat szinte megegyezik, és a  $\mathcal{S}$  útvonalválasztása jóval egyszerűbb.

#### 4.4. Összefoglaló

Ebben a fejezetben leírt keretrendszerrel használva megvizsgáltam egy valós hálózatot az openflights repülési adatbázisát felhasználva. A szimulációs keretrendszerben meghatározott lépésekkel elvégeztem, a szükséges módosításokat megtettem, ahol a probléma miatt erre szükség volt. A repülési adatbázis alapján létrehoztam az irányított, vektor-súlyozott gráfot, az élsúlyokat a vizsgálandó algebrák alapján számítottam ki és rendeltem az élekhez.

A kész tervet a saját magam írt szimulátor szoftverrel hajtottam végre, az eredményeket közönséges táblázatkezelővel elemeztem. Megvizsgáltam a Shortest ( $\mathcal{S}$ ), a LeastHop ( $\mathcal{L}$ ), az Összekötő-keresés ( $\mathcal{O}$ ) és a Korai-elfogadó-keresés ( $\mathcal{K}$ ) algebrákat. Mivel az utób-

bi kettő nem jól viselkedő algebra, ezeket csak a Shortest és a LeastHop algebrákkal való lexikografikus szorzatukban vizsgáltam. Az Összekötő-keresés-LeastHop ( $\mathcal{OL}$ ) és a Korai-elfogadó-keresés-LeastHop ( $\mathcal{KL}$ ) algebrákat, valamint a Shortest-LeastHop ( $\mathcal{SL}$ ) és LeastHop-Shortest ( $\mathcal{LS}$ ) algebrákat nem kellett vizsgálnom. Az első kettőnek elméleti megfontolások alapján nincs új információtartalma, utóbbi kettő pedig előáll a külön-külön szimulált algebrák speciális alakjaként.

A korábban definiált pontrendszer felhasználva beazonosítottam a repülési adatok alapján legtöbb javítási lehetőséget nyújtó algebrát, a Shortest algebrát. Ez elég pontosan becsli az eredetei megfigyelt hálózatot, ugyanakkor megmutat olyan lehetőségeket, amik mentén javíthatunk a jelenlegi útvonalválasztáson. Emellett rávilágítottam a tényre, hogy az  $\mathcal{OS}$  algebrát akár használhatják is – elég jó becslése az eredeti hálózatnak – viszont a  $\mathcal{S}$  algebra egyszerűbb és pontosabb becslés, azaz erre le lehetne cserélni a jelenleg alkalmazott szabályrendszert.

## 5. fejezet

# Összefoglalás

A feladatkiírás pontjainak eleget tettem. A szakirodalom alapján a hálózatkutatás eredményeit áttekintettem, különös tekintettel az útvonalválasztás matematikai kérdésére. Megvizsgáltam több különböző, valós életből származó hálózati útvonalválasztással szoros kapcsolatban álló problémakört, és mindegyik ilyenhez definiáltam útvonalválasztási stratégiákat, melyek vizsgálata új megvilágításba helyezné ezeket a problémákat. Specifikáltam egy szimulációs keretrendszert, mellyel lehetőség nyílik az új stratégiák vizsgálatára. Egy ilyen vizsgálatot el is végeztem, amely során a repülőtársaságok útvonalválasztását vizsgáltam meg.

Az 1. fejezetben áttekintettem a szakirodalmat, összeszedve a legfontosabb állomásokat. Rámutattam, hogy az Internet AS-szintű topológiáján kívül, a más tudományterületekről származó problémák útvonalválasztásáról sem tudunk sok minden és ezért szükséges egy olyan eszköz, ami a policy-feltárás feladatát – általános esetben is – hatékonyan el tudja látni. Ehhez definiáltam a *routing algebrát*, bemutattam a legfontosabb tulajdonságait és műveleteit, emellett a legszélesebb körben használt policy-k algebráit is ismertettem.

A 2. fejezetben megvizsgáltam a hálózatkutatás szempontjából alapvető modellekét, melyeket a lokális- vagy globális optimalizálás és a közös- vagy egyéni érdekek követése tulajdonoságok alapján karakterizáltam. Bemutattam a fertőző betegségek vizsgálatára használt matematikai modellt, megvizsgáltam a téma kör útvonalválasztási kérdéseit és kijelöltettem a két, a problémakort jól jellemző policy-t. Emellett rávilágított a trend- és a vírusterjedés útvonalválasztási szempontbeli hasonlóságaira és különbségeire és definiáltam két új trendterjedési policy-t.

Megvizsgáltam a már ismertetett policy-kon (ld. az 1.5. alfejezet) kívül az Internet tartományszintű gráfjának az alapszabályát, a völgymentességet, illetve felvázoltam a hiperbolikus térbe ágyazás – általa pedig az elakadásmentes mohó útvonalválasztás – lehetőségét.

A 3. fejezetben specifikáltam és megterveztem egy szimulációs keretrendszert, mellyel valós hálózati problémákon lehet tesztelni különböző útvonalválasztási stratégiák alapján leírt algebrákat. Definiáltam a szimulációhoz szükséges előfeldolgozási, adattisztítási lépé-

seket és magát a szimulációs folyamatot. Emellett kidolgoztam egy pontrendszeret, mellyel értékelni lehet a szimulációs eredményeket. Referencia szimulációkkal úgy alakítottam ki ezt a pontrendszeret, hogy ne csak egy relatív skálát kapjak, ahol csak a vizsgált algebrákat hasonlíthatom össze egymással, hanem egy abszolút mércét is jelentsen az eredmény pontértéke.

A 4. fejezetben bemutattam a vizsgálandó valós hálózatot. Az adatok feldolgozása után meghatároztam a vizsgálandó hálózatot. Definiáltam a vizsgálandó metrikákat, amik minden össze tudtam hasonlítani a valós- és a policy vezérelt útvonalválasztás által meghatározott útvonalakat és hálózatokat. minden pontpárra megvizsgálta, az eredeti és a szimulált út különbségeit (távolság, lépésszám). A valós és a szimulált hálózatok statisztikai összehasonlítását is elvégeztem, melyben figyelmet fordítottam a fokszámeloszlás, a hálózatok átmérőjének és az élösszefüggésének összehasonlítására.

A saját magam írt szoftverrel futtatott szimulációk eredményeinek alapján kijelenthető, hogy a diplomamunkámban tárgyalt keretrendszer – megfelelő körülmények között – alkalmas valós hálózati útvonalválasztási problémák vizsgálatára, algebrák tesztelésére.

# Ábrák jegyzéke

2.1.	A BGP egyszerűsített képe és a völgymentesség szerinti irányítás. . . . .	25
2.2.	A hiperbolikus sík Poincaré-féle diszk modellje és a felső félsík modellje [15].	25
4.1.	Az OpenFights adatbázisában szereplő repterek. . . . .	38
4.2.	Az OpenFights adatbázisában szereplő repülési útvonalak. . . . .	39
4.3.	A $\mathcal{S}$ algebra által generált gráf fokszámeloszlása. . . . .	43
4.4.	A $\mathcal{S}$ algebra által generált gráf csomópontonkénti átlagfokszám-eloszlása. .	43
4.5.	A $\mathcal{L}$ algebra által generált gráf fokszámeloszlása. . . . .	44
4.6.	A $\mathcal{L}$ algebra által generált gráf csomópontonkénti átlagfokszám-eloszlása. .	44
4.7.	Az $\mathcal{OS}$ algebra által generált gráf fokszámeloszlása. . . . .	45
4.8.	Az $\mathcal{OS}$ algebra által generált gráf csomópontonkénti átlagfokszám-eloszlása.	46
4.9.	A $\mathcal{KS}$ algebra által generált gráf fokszámeloszlása. . . . .	46
4.10.	A $\mathcal{KS}$ algebra által generált gráf csomópontonkénti átlagfokszám-eloszlása.	47

# Irodalomjegyzék

- [1] George Apostolopoulos, Roch Guérin, Sanjay Kamat, and Satish K. Tripathi. Quality of service based routing: A performance perspective. *SIGCOMM Comput. Commun. Rev.*, 28(4):17–28, October 1998.
- [2] S. Blower and D. Bernoulli. An attempt at a new analysis of the mortality caused by smallpox and of the advantages of inoculation to prevent it. 1766. *Reviews in Medical Virology*, 14(5):275–288, October 2004.
- [3] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, DIALM '99, pages 48–55. ACM, 1999.
- [4] Hyunseok Chang, Sugih Jamin, and Walter Willinger. Inferring as-level internet topology from router-level path traces. In *In Proceedings of SPIE ITCom 2001*, 2001.
- [5] Zoltán L. Csaba and Judit Pál. Az utálat és az iskolai agresszió empirikus hálózat-elemzése. VII. HUNNET konferencia. Budapesti Corvinus Egyetem, 2010.
- [6] D. J. Daley, J. Gani, and J. M. Gani. *Epidemic Modelling: An Introduction*. Cambridge Studies in Mathematical Biology. Cambridge University Press, 2001.
- [7] Gábor Döbrei. Gráf-beágyazási technikák kommunikációs hálózatokban. Bachelor's thesis, Budapest University of Technology and Economics, Hungary, February 2012.
- [8] Gábor Döbrei. Nagyméretű hálózatok evolúciója - trendterjedés vizsgálata szimulációs eszközökkel. Project Laboratory, Budapest University of Technology and Economics, Hungary, February 2014.
- [9] D. Evans. The internet of things - how the next evolution of the internet is changing everything. Cisco Internet Business Solutions Group (IBSG), April 2011.
- [10] Malcolm Gladwell. *Fordulópont: Ahol a kis különbségekből nagy változás lesz*. HVG Kiadói Rt., 2007.
- [11] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. IEEE, volume 3, pages 1371–1380, 2000.

- [12] Timothy G. Griffin and João Luís Sobrinho. Metarouting. *SIGCOMM Comput. Commun. Rev.*, 35(4):1–12, August 2005.
- [13] A.J.T. Gurney and T.G. Griffin. Lexicographic products in metarouting. In *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*, pages 113–122, October 2007.
- [14] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721, August 1927.
- [15] R. Kleinberg. Geographic routing using hyperbolic space. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1902–1909, 2007.
- [16] Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. In *Network Protocols, 1997. Proceedings., 1997 International Conference on*, pages 191–202, October 1997.
- [17] Hiroshi Nagamochi and Toshihide Ibaraki. Augmenting edge-connectivity over the entire range in  $O(nm)$  time. *J. Algorithms*, 30:253–301, 1996.
- [18] Gábor Rétvári, András Gulyás, Zalán Heszberger, Márton Csernai, and József J Bíró. Compact policy routing. *Distributed computing*, 26(5-6):309–320, 2013.
- [19] João Luís Sobrinho. Algebra and algorithms for QoS path computation and hop-by-hop routing in the internet. *Networking, IEEE/ACM Transactions on*, 10(4):541–550, August 2002.
- [20] João Luís Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '03*, pages 49–60, 2003.
- [21] Hongsuda Tangmunarunkit, Ramesh Govindan, Scott Shenker, and Deborah Estrin. The impact of routing policy on internet paths. In *in Proc. 20th IEEE INFOCOM*, pages 736–742, 2001.
- [22] William P. Thurston and Silvio Levy. *Three-dimensional Geometry and Topology*. Princeton University Press, 1997.
- [23] Feng Wang and Lixin Gao. On inferring and characterizing internet routing policies. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement, IMC '03*, pages 15–26. ACM, 2003.
- [24] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *Selected Areas in Communications, IEEE Journal on*, 14(7):1228–1234, September 1996.

- [25] Duncan J. Watts. Networks, dynamics, and the small-world phenomenon. *American Journal of Sociology*, 105:493–527, 1999.

# Függelék

## F.1. Számításelméleti alapok

**F.1.1. Definíció ( $O$ ,  $\Omega$ ,  $\Theta$ ).**  $f, g : \mathbb{N} \rightarrow \mathbb{R}$

- Ordó:  $f(n) = O(g(n))$ , ha  $\exists c > 0, n_0 > 0 : |f(n)| \leq c|g(n)|, \forall n > n_0$ .
- Omega:  $f(n) = \Omega(g(n))$ , ha  $\exists c > 0, n_0 > 0 : |f(n)| \geq c|g(n)|, \forall n > n_0$
- Teta:  $f(n) = \Theta(g(n))$ , ha  $f(n) = O(g(n))$  és  $f(n) = \Omega(g(n))$ , azaz  $\exists c_1, c_2 > 0, n_0 > 0 : c_1|g(n)| \leq |f(n)| \leq c_2|g(n)|, \forall n > n_0$

**F.1.1. Megjegyzés.** A nagy ordós jelölésből szokás kihagyni a  $\log n$ -es szorzót, minthogy egy decimális számot mindenkorban át kell konvertálni  $\log n$  bitre, így egy algoritmus lépésszámánál vagy (lokális / globális) memóriaigénynél biztosan nem tudjuk megspórolni:  $\tilde{O}(n) = O(n \log n)$ .

**F.1.2. Definíció (K-szorosan pont-összefüggő gráf 1.).** Egy  $G$  gráf  $k$ -szorosan pont-összefüggő, ha tetszőleges  $k$ -nál kisebb elemszámú csúcshalmazát elhagyva összefüggő gráfot kapunk, és  $|V(G)| \geq k$ .

**F.1.3. Definíció (K-szorosan él-összefüggő gráf 1.).** Egy  $G$  gráf  $k$ -szorosan él-összefüggő, ha tetszőleges  $k$ -nál kisebb elemszámú élhalmazát elhagyva összefüggő gráfot kapunk.

Két alternatív definíció:

**F.1.4. Definíció (K-szorosan pont-összefüggő gráf 2.).** Egy  $G$  gráf  $k$ -szorosan pont-összefüggő, ha tetszőleges  $k$ -nál kisebb elemszámú csúcshalmazát elhagyva összefüggő gráfot kapunk, és  $|V(G)| \geq k$ .

**F.1.5. Definíció (K-szorosan él-összefüggő gráf 2.).** Egy  $G$  gráf  $k$ -szorosan él-összefüggő, ha tetszőleges  $k$ -nál kisebb elemszámú élhalmazát elhagyva összefüggő gráfot kapunk.

**F.1.6. Definíció (Összefüggő gráf átmérője).** A  $G$  összefüggő gráf átmérője a két legtávolabbi csúcsának távolsága, azaz az összes csúcspár közötti legrövidebb utak közül a leghosszabbnak a hossza. Jelölje  $P(u, v)$  az  $u$  és  $v$  csúcsok közötti utak halmazát, és  $l(p)$  a  $p$  út hosszát, ekkor a gráf átmérője:

$$D_G = \max_{u, v \in V} \min_{p \in P(u, v)} l(p)$$

**F.1.7. Definíció (K-reguláris gráf).** Egy irányítatlan  $G = (V, E)$  gráfot  $k$ -regulárisnak nevezünk, ha minden csúcsa pontosan  $k$ -fokú:  $d(v) = k$ ,  $\forall v \in V$ .

**F.1.8. Definíció (Síkba rajzolható gráf).** Egy gráf síkba rajzolható, ha a pontjainak kölcsönösen egyértelműen megfeleltethetők síkbeli pontok, az éleknek pedig a megfelelő csúcsokat összekötő egyenes szakaszok úgy, hogy a különböző szakaszok legfeljebb csak a végpontjaikban találkozhatnak.

**F.1.9. Definíció (DAG).** Egy irányított gráf akkor DAG (Directed Acyclic Graph), ha nem tartalmaz irányított kört.

**F.1.10. Definíció (Topologikus rendezés).** A  $G = (V, E)$  egy irányított gráf.  $G$  egy topologikus rendezése, a csúcsok egy olyan  $v_1, v_2, \dots, v_{|V|}$  sorrendje, melyben  $i \rightarrow j \in E$  esetén  $v_i$  előbb van, mint  $v_j$ .

**F.1.11. Tétel (Topologikus rendezés létezése).** Egy irányított gráfnak akkor és csak akkor van topologikus rendezése, ha DAG.

**F.1.12. Tétel (Maximális út létezése).** Egy irányított, súlyozott gráfban akkor lehet maximális út  $u$  és  $v$  között, ha a gráf DAG.

**F.1.13. Definíció (Gráfok izomorfiája).** A gráfizomorfizmusok gráfok közötti bijektív struktúratartó leképezések, értve ezalatt azt, hogy a függvény és az inverz függvény egymárt szomszédos csúcsokat szomszédos csúcsokra képezzék: Adott  $G(V, E)$  és  $G'(V', E')$  gráfok. Egy  $f : V \rightarrow V'$  bijektív függvény gráfizomorfizmus, ha  $\{u, v\} \in E \Leftrightarrow \{f(u), f(v)\} \in E'$ . Ilyenkor azt mondjuk, hogy  $G(V, E)$  és  $G'(V', E')$  izomorf.

**F.1.14. Definíció (Eldöntési probléma).** Egy eldöntési problémához tartozó  $L$  nyelv azoknak a bemeneteknek a halmaza, amelyekre a válasz IGEN. A lehetséges bemeneteket (amik tehát vagy beletartoznak  $L$ -be vagy nem), szavaknak hívjuk. Egy  $X$  eldöntési probléma és  $x$  bemenet esetén  $x \in X$  jelöli, hogy az  $x$  bemenetre a válasz IGEN.

## F.2. Az útvonalválasztás teljes modellje

Ahhoz, hogy felépíthessük a routing teljes modelljét, a routing algebrákon (az 1.2. rész) kívül egy routing függvényre van szükség. Ebben a modellben a csomagok (ahogyan a valóságban is) hasznos teherből (payload) és egy header-ből állnak. Ha adott az  $\mathcal{A}$  routing algebra és a  $G$  gráf, akkor a policy routing függvény az  $R : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N} \times \mathbb{N}$  leképezés, a csúcsok  $L_V : V \mapsto \mathbb{N}$  címkézésével és az élek  $L_E : E \mapsto \mathbb{N}$  címkézésével, a következőképpen: minden  $s, t$  pontpárra egymás után alkalmazva R-t:

$$(h_{i+1}, l_{i+1}) = R(v_i, h_i), \forall i = 1, \dots, k-1$$

megadja a preferált  $p_{st}^* = (s =)v_1, v_2, \dots, v_k (=t)$  utat  $\mathcal{A}$  szerint a megfelelő  $l_{i+1} = (v_i, v_{i+1})$  él-címkékkal, ahol  $h_1$  egy alkalmas kezdő header. Azt mondjuk, hogy  $R$  megvalósítja az  $\mathcal{A}$  policy-t  $G$ -n. Még néhány óvintézkedést meg kell tennünk, hogy a címkékkel nehogy több routing információt kelljen kódolni a szükségesnél, azaz *clogn*<sup>1</sup>, valamilyen alkalmas  $c$  konstanssal.

Ezek alapján tehát a routing a következőképpen történik: Ha egy  $u$  csomópont kap egy üzenetet  $h$  header-rel, akkor egyszerűen kiszámolja az  $R_u(h)$  lokális routing függvény értékét:  $R_u(h) = R(u, h)$ , hogy megkapja az új header-t,  $h'$ -t és a kimenő portot,  $l$ -t. Ezután  $u$  beállítja a csomag header-jének  $h'$ -t és továbbküldi  $l$ -en keresztül. A routing függvénytel már könnyen megadható egy hálózat minden csomópontjának a lokális memóriaigénye ahoz, hogy egy adott routing policy-t valósítson meg:

**F.2.1. Definíció (Routing policy megvalósításához szükséges lokális memória).**

Az  $\mathcal{A}$  routing policy megvalósításához szükséges  $\mathcal{M}_{\mathcal{A}}$  lokális memória:

$$\mathcal{M}_{\mathcal{A}} = \max_{G \in \mathcal{G}_n} \min_{R \in \mathcal{R}} \max_{u \in V} \mathcal{M}_{\mathcal{A}}(R, u),$$

ahol  $\mathcal{M}_{\mathcal{A}}(R, u)$  az  $R_u$  lokális routing függvény kódolásához szükséges bitek minimális száma,  $\mathcal{R}$  azon policy routing függvények halmaza, melyek megvalósítják  $\mathcal{A}$  policy-t valamely  $G$  gráfon, és  $\mathcal{G}_n$  az összes  $n$  csúcsú gráfok halmaza.

## F.3. Tételek a routing algebrák témaakörében

**F.3.1. Tétel (Tömöríthetőség 1).** Ha  $\mathcal{A}$  algebra kiválasztó ( $S$ ) és monoton ( $M$ ), akkor tömöríthető.

**F.3.2. Tétel (Tömöríthetőség 2).** Ha  $\mathcal{A}$  algebra szigorúan monoton ( $SM$ ), akkor nem tömöríthető.

Ennél egy általánosabb téTEL, aminek következménye az előző téTEL:

**F.3.3. Tétel (Tömöríthetőség 3).** Ha  $\mathcal{A}$  algebra tartalmaz egy delimitált ( $D$ ), szigorúan monoton ( $SM$ ) subalgebrát, akkor  $\mathcal{A}$  nem tömöríthető.

---

<sup>1</sup> Ez a címek kódolásához szükséges információ.

#### F.4. A szimulátor technikai részletei

A szimulátort Java nyelven fejlesztettem, mivel semmilyen megkötés nem volt a a 3. fejezet specifikációjában, ami alapján ne lett volna alkalmas ez a megoldás. Azért, hogy ne csak a szimulációs eredményeket tudjam felhasználni a diplomamunkában, illetve ne csak egy elméleti megoldás legyen a keretrendszerem, törekedtem a szép, tiszta, érthető kód írására. Így mellékletként ez is értékelhető, a további munka során felhasználható.

A fejlesztés során kér külső függvénykönyvtárat használtam fel: a Google Guava<sup>2</sup> 18-as verzióját az adatszerkezetek és bejárásukra, illetve a Google Gson<sup>3</sup> 2.2.4-es verzióját az eredmények JSON formátumban való mentéséhez és visszaállításukhoz.

A szimulátor alapvetően három részre tagolódik: a hálózatot leíró osztályok, az algebrákat leíró osztályok és a szimulációt futtató osztályok.

A hu/dobrei/diploma/network package-ben vannak az adott feladathoz tartozó hálózatot leíró osztályok, jelen esetben az Airline, Airport, Flight, Route és az OpenFlightsNetwork osztályok.

A hu/dobrei/diploma/algebra package-ben van az AbstractAlgebra absztrakt osztály és a belőle származó BusyAirportFindingAlgebra, LeastHopAlgebra, EarlyAdopterFindingAlgebra és a ShortestAlgebra osztályok.

A harmadik rész a hu/dobrei/diploma/routing package-ben levő osztályok, melyek a szimulációért, a végeredmények elemzéséért és az eredmények fájlba mentéséért felelősek.

A tervezéstől kezdve a fejlesztése során végig arra törekedtem, hogy moduláris, könnyen újrahasználható szoftvert fejlesszek, melyet később tovább lehet adni. Ennek érdekében a lehető legtisztább, legegyszerűbb megoldásokat alkalmaztam, illetve ahol csak tudtam, ragaszkodtam a clean code<sup>4</sup> elveihez.

Összefoglalva elmondható, hogy sikerült elérni a célt a saját fejlesztésű szimulátorral: nem kellett másik szoftver használatát megtanulnom, teljesen biztos lehetek benne, hogy az útvonalválasztási algoritmus optimálisan van megírva és az algebrák implementálása pontos, egyszerű, de ésszerű. Emellett további speciális felhasználásra viszonylag egyszerűen lehet módosítani a meglévő kódbázist.

---

<sup>2</sup>A Google Guava forrása: <https://github.com/google/guava>

<sup>3</sup>A Google Gson forrása: <https://code.google.com/p/google-gson/>

<sup>4</sup>Robert C. Martin: Clean Code – A Handbook of Agile Software Craftsmanship