



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Informatikai Tudományok Doktori Iskola
Távközlési és Médiainformatikai Tanszék

OPTIMÁLIS ERŐFORRÁS-MEGOSZTÁS HAGYOMÁNYOS
EGYENLETESEN OSZTÓ TERHELÉSKIEGYENLÍTŐ
RENDSZEREK HASZNÁLATÁVAL

Németh Krisztián

okl. mérnökinformatikus

Tézisfüzet

Témavezető:

Dr. Rétvári Gábor
tudományos főmunkatárs

Budapest

2018

1. Bevezetés

A különálló erőforrások együtt történő kezelése és a felhasználók közötti megosztása az Internet egyik alapvető jellemzője. Ez a módszer, amelyet gyakran erőforrás-megosztási elvnek (Resource Pooling Principle [1]) neveznek, a mai kommunikációs hálózatokban sok helyen megfigyelhető. Erre az elvre példa többek között a többutas útvonalválasztás (multipath routing), a többutas hálózatelérés (multihoming), a terheléselosztás alkalmazásrétgbeli szerverek – például web- vagy adatbázisszerverek – között, a forgalomelvezetésben (Traffic Engineering¹) alkalmazott terheléskiegyenlítés, valamint az Ethernet linkaggregációs csoportok (Link Aggregation Groups). A tartalomkézbesítő hálózatok (Content Delivery Network) szintén az erőforrás-megosztási elvet alkalmazzák, akárcsak a felhőalapú tárolási és a felhőalapú számítási rendszerek. E szolgáltatások megvalósítására adatközpontokat építenek, melyekben gyakran alkalmaznak párhuzamos útvonalakat, amelyek azonban az esetek jelentős részében aszimmetrikus kapacitásúak [2]. Ezen felül is számos új koncepció jelent meg az elmúlt években, mint például a hálózatvirtualizáció vagy a szoftveralapú hálózatok (Software Defined Networking, SDN), melyek szintén megosztást használnak a hálózati erőforrások optimális kihasználása érdekében.

Bár ez a lista messze nem teljes, mégis jól mutatja, hogy mennyire különböző területeken alkalmazható az erőforrások osztott használata. Több okból is célszerű ezt az elvet követni. Először is, a vele járó redundancia növeli az ellenálló képességet a komponensek hibáival szemben. Másodszor, egy ideiglenesen megnövekedett igényhez dinamikusan több erőforrást rendelve magasabb szintű szolgáltatásokat lehet nyújtani változatlan infrastruktúrával a statisztikus nyalábolás előnyeit kihasználva. Harmadszor, a nagyobb szabadság az igények és erőforrások összerendelésében hatékonyabb hálózatfelhasználást és jobban skálázható szolgáltatásokat tesz elérhetővé.

Az erőforrás-megosztás megvalósítása azonban nem mindig könnyű, hiszen a terheléselosztók gyakran csak azonos arányban képesek szétosztani a bejövő forgalmat az erőforrások között. Például egy terheléselosztó, melyhez két webszerver tartozik, tipikusan megfelezi közöttük a bejövő igényeket. Ez azonban jelentősen visszaveti a rendszer összeteljesítményét, ha mondjuk az egyik szerver kétszer olyan erős, mint a másik. Hasonlóképpen az olyan útvonalválasztó protokollokban, mint az OSPF [3] vagy az IS-IS [4], az ún. Equal-Cost Multipath-t (egyenlő költségű többutas terjesztés, továbbiakban csak ECMP) használják a forgalom elosztására az azonos költségű – avagy azonos hosszúságú, ami ugyanazt jelenti ez esetben – legrövidebb utak között. Az ECMP azonban csak egyenletesen képes elosztani a forgalmat ezen útvonalak között, akkor is, ha az utak kapacitása különbözik, ami komoly akadályokat gördít az optimális forgalomelvezetés útjába [2, 5, 6, 7].

Sok meglévő javaslat létezik, melyek e problémának egyes speciális eseteivel foglalkoznak. Például a Weighted Cost Multipathing (súlyozott útvonalú többutas terjesztés, WCMP [2]) adatközpontokban történő nem egyenletes megosztásra tesz javaslatot. SDN-képes kapcsolókat tételez fel, és a működése szabálytábla-bejegyzések többszörözésén alapszik. A Niagara [6] egy másik SDN-en alapuló javaslat, amely rugalmas forgalom-szétosztást biztosít

¹Az angol „Traffic Engineering” kifejezést többféleképpen is szokták magyarra fordítani, például forgalomtervezésnek, forgalomterelésnek vagy forgalomelvezetésnek. A „forgalomtervezés” hűbb az angol eredetihez, míg az utóbbi kettő közelebb áll a kifejezés jelentéséhez. Tézisfüzetemben végig a „forgalomelvezetés” fordítást fogom használni.

terheléelosztók között. Az általa létrehozott SDN szabályokban a forráscímek utolsó biteit használja fel. A Fibbing [8] egy újabb érdekes architektúra, amely SDN nélkül ígér központi irányítást elosztott útvonalválasztás felett. Működése során nem létező csomópontokat és linkeket hirdet szabványos OSPF protokollüzenetekben. A Fibbingnek egy újabb alkalmazása éppen a terheléelosztást kívánja megvalósítani [9]. Mindezek a javaslatok azonban többé-kevésbé egyetlen alkalmazási területre koncentrálnak, és vagy olyan technológián alapszanak, amely jelenleg még nem széleskörűen elterjedt (pl. SDN), vagy, mint a Fibbing esetében, egy új absztrakciós szintet vezetnének be, és nem világos még, hogy a szolgáltatók vállalnák-e az ezzel kapcsolatos többletfeladatokat.

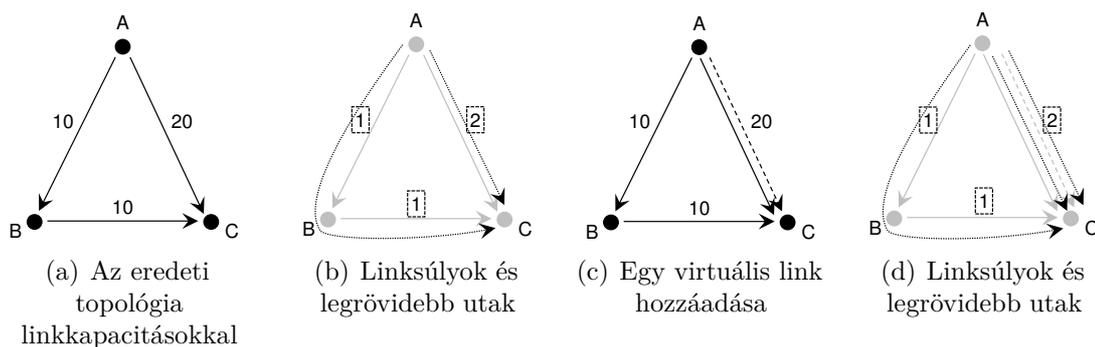
A problémára egy új megoldást javaslok, amelyet *virtuális erőforrás-foglalásnak* (Virtual Resource Allocation) nevezek, és amelyre a továbbiakban az angol rövidítésével VRA-ként hivatkozom. A javaslatom célja optimális erőforrás-megosztás létrehozása hagyományos egyenletesen osztó terheléskiegyenlítő rendszerek használatával. A VRA alapötlete az, hogy virtuálisan megtöbbszörözzük az elérhető párhuzamos erőforrásokat, és így a terheléelosztó rendszer nagyobb számú erőforrást lát, mint amennyi valójában létezik. Ezután a virtuális erőforrásokat kisebb csoportokba gyűjtjük és a fizikai erőforrásokhoz rendeljük. Ilyen módon „becsapjuk” a hagyományos egyenletesen osztó technológiát annak érdekében, hogy megközelíthessük a megkívánt nem egyenletes terheléelosztást a meglévő erőforrásokat használva.

Folytatva az előző példát, tegyük fel, hogy valaki két virtuális gépet telepít az erősebb webszerverre, és a változatlanul hagyott gyengébb szerverrel együtt a terheléelosztóba be-regisztrálja ezeket. Ekkor az elosztó három szervert fog látni, és megpróbálja elharmadolni közöttük a teljes terhelést, ami azt jelenti, hogy a nagyobb kapacitású fizikai szerver az összes beérkező igény kétharmad részét kapja meg, éppen ahogyan azt szeretnénk volna.

A virtuális erőforrás-foglalás bemutatására a következő példát a forgalomelvezetés témaköréből választottam. Tekintsük az 1(a) ábrán látható háromszög alakú hálózatot! Tegyük fel, hogy 30 egységnyi forgalmat szeretnénk eljuttatni A -ból C -be úgy, hogy egyik linket sem terheljük túl. Hagyományos OSPF-et használva lehetőségünk van a linksúlyok² beállítására, így könnyen készíthetünk két egyenlő összsúlyú legrövidebb utat (vagyis minimális összsúlyú/összköltségű útvonalakat): $A - B - C$ -t és $A - C$ -t. Egy lehetséges megfelelő súlykiosztást mutat az 1(b) ábra. Ugyanakkor azonban az OSPF ECMP csak egyenletes forgalomelosztást tesz lehetővé a legrövidebb utak között, ami 150%-os terhelést eredményez az $A - B$ és $B - C$ linkeken.

Ha viszont létre tudunk hozni egy virtuális linket a meglévő $A - C$ link felett, és azt is kiajánljuk az OSPF-nek (ld. az 1(c) ábrát), az készségesen háromfelé osztja a forgalmat, amelynek egyharmada az $A - B - C$ útvonalon halad, a maradék pedig az $A - C$ fizikai linken (1(d) ábra). Természetesen az $A - C$ fizikai link kapacitása nem változik attól, hogy létrehozunk rajta egy virtuális linket, ez pusztán a link jobb kihasználását teszi lehetővé OSPF környezetben. A linksúlyok is változatlanok maradnak, és az új virtuális link súlya megegyezik a hozzá tartozó fizikai link súlyával. Ezzel az egyszerű adminisztratív beavatkozással már el tudjuk vezetni a forgalmat a példahálózatunkon anélkül, hogy bármelyik linket is túlterhelnénk.

²A linksúlyokat gyakran linkköltségeknek, linkmetrikáknak, vagy SPF metrikáknak (SPF: Shortest Path First, egyfajta útvonalválasztó elv) is nevezik. E tézisfüzetben, mint általában a szakirodalom is, e kifejezéseket egyenértékűként használom.



1. ábra. Egy háromszög alakú példahálózat. Igény: $A \rightarrow C : 30$

A VRA javaslatom egyik előnye, hogy fokozatosan is bevezethető, hiszen semmi akadálya annak, hogy a hálózati csomópontoknak csak egy részében hozzunk létre virtuális erőforrásokat. Továbbá – más javaslatokkal ellentétben – a VRA teljesen kompatibilis a hálózatok meglévő szoftver- és hardverkomponenseivel. Végül a VRA rendkívül hatékony a numerikus eredményeim szerint, melyek azt mutatják, hogy pusztán egy kis számú virtuális erőforrás hozzáadásával az ideális osztásarány nagyon jól megközelíthető, ami jelentős teljesítménynövekedést eredményez.

2. Kutatási célkitűzések

Munkámban az IP hálózatokban történő forgalomelvezetés [10] példáján keresztül mutatom be a VRA javaslatot. Az alapötlet az, hogy hozzunk létre virtuális linkeket a meglévő fizikaiak felett és ajánljuk fel ezeket is az OSPF-nek, ami által közel optimális forgalomelvezetés valósítható meg anélkül, hogy módosítani kellene a hálózati infrastruktúra szoftverén vagy hardverén. Hadd emeljem azonban ki, hogy a forgalomelvezetés csak egy példaalkalmazás a VRA koncepcióra, amelynek a lehetséges alkalmazási területe ennél jóval szélesebb. Egy másik példa felhasználási lehetőségként megemlítem, hogy bizonyos SDN-alapú rendszerekben a VRA eredményesen használható a szabálytáblázatok optimalizálására.

Sokféleképpen lehetséges közel optimális forgalomelvezetést biztosítani, amelyek közül a továbbiakban én hárommal foglalkozok. Az elsőben, melyet *átfedő optimalizálásnak* (Overlay Optimization) nevezek, lehetőség van végponttól végpontig tartó alagutak létrehozására, például MPLS-TE [10] segítségével. Azt is felteszem továbbá, hogy a jobb erőforrás-kihasználás érdekében több alagút is létezhet egyetlen forrás- és célcsomópontpár között. A VRA koncepciót alkalmazva ezeket az alagutakat virtuálisan többszörözhetjük, majd OSPF ECMP-t használhatunk ezen átfedő hálózat felett. A cél ez esetben az, hogy megtaláljuk egy adott csomópontpárt összekötő útvonalak között előre definiált forgalomosztás-arány legjobb közelítését, úgy, hogy csak korlátos számú virtuális útvonalat használunk.

A másik kettő javasolt forgalomelvezetési módszerben, melyeket *helyi egyenrangú optimalizálásnak* (Peer-Local Optimization) és *globális egyenrangú optimalizálásnak* (Peer-Global Optimization) nevezek, nincsenek átfedő hálózatok, így a forgalomelvezetés optimalizálása

közvetlenül a fizikai infrastruktúrán történik. Továbbá semmiféle kifinomult forgalomelvezetési módszer jelenlétét sem tételezem fel, csupán az OSPF-et használjuk ilyen célokra. Ezt OSPF forgalomelvezetésnek (OSPF Traffic Engineering, a továbbiakban röviden OSPF-TE) nevezik és az alapötlete az, hogy az adminisztratív linkköltségeket úgy állítjuk be, hogy az OSPF által kiszámított legrövidebb utak pontosan az adminisztrátor által kiválasztottakkal egyezzenek meg [11, 12].

Kiegészítettem a hagyományos OSPF-TE koncepciót azzal, hogy virtuális linkeket készítek a meglévő fizikai linkekkel párhuzamosan.³ A megoldandó mérnöki probléma ez esetben az, hogy egy hálózati tartományban minden fizikai linkhez meghatározzuk a költséget és a párhuzamos virtuális linkek számát, úgy, hogy ezzel egy előre definiált mérőszámot optimalizáljunk. Ahogy a neve is jelzi, a helyi egyenrangú optimalizálás esetében ezeket a döntéseket helyileg, a csomópontok szintjén hozzuk meg, míg a globális egyenrangú optimalizálás esetében ez központosítva, hálózati szinten történik. Munkámban az alkalmazott mérőszám a széleskörűen elfogadott maximális linkkihasználtság, melyet minimalizálni szeretnénk a tartomány összes linkje felett. A linkkihasználtságot a szokásos módon értelmezzük: a linken lévő forgalom mennyiség osztva a link kapacitásával.

A probléma nem egyszerű, hiszen ismert, hogy önmagában a legjobb linksúlykonfiguráció megtalálása a (virtuális linkek nélküli) OSPF-TE-hez is NP-nehéz [5], és gyors algoritmussal semmilyen konstans szorzó erejéig nem lehetséges az optimumot még közelíteni sem [13].

3. Kutatási módszertan

Az analitikus, matematikai vizsgálatot választottam a kutatási célkitűzéseim eléréséhez eszközül. Azért döntöttem így, mert az ily módon elért eredmények általában univerzálisabbak és megalapozottabbak, mint azok, amelyekhez szimulációk vagy mérések útján jutunk. Az analitikus módszer egyik hátránya az, hogy a problémáknak egy jelentős része túl bonyolult az ilyen módon történő leíráshoz. Jelen esetben azonban ez nem jelentett leküzdhetetlen akadályt, így tézisállitásaim jelentős része saját magam által kimondott matematikai tételeken és azok precíz bizonyításain alapszik.

Néhány tézisemben meglévő problémák megoldására javaslok algoritmusokat. Ezen esetekben az algoritmusok számítási komplexitását azok analitikus vizsgálatával határoztam meg. Sok általam javasolt algoritmus lineáris program vagy (kevert) egész értékű lineáris program megoldásán alapszik. Vannak téziseim, melyek NP-teljességi bizonyításokat tartalmaznak. Ezekben a szokásos módon Karp-redukciót alkalmazok.

Készítettem továbbá egy szimulációs rendszert annak érdekében, hogy megvizsgálhassam, hogy a javasolt algoritmusok hogyan teljesítenek valóságghú környezetekben. E szimulátorban az algoritmusokat összemértem egymással, illetve a leginkább elfogadott jelenlegi módszerrel (TOTEM [14, 15]) is. A szimulátort C++ nyelven írtam, felhasználva a LEMON Graph Library [16] könyvtárat. A beágyazott lineáris programokat az IBM ILOG CPLEX Optimizer [17] segítségével oldottam meg. Az ily módon nyert eredmények értékes információkat szolgáltatnak az általam javasolt optimalizációs technikákban rejlő lehetőségekről.

³Sokféleképpen lehetséges virtuális linkeket létrehozni, de ennek pontos módszere kívül esik munkám hatókörén, mivel én most csak a virtuális linkeknek a hálózat teljesítőképességére való hatásával foglalkozok.

4. Új eredmények

4.1. Virtuális erőforrás-foglalás és átfedő optimalizálás

1. téziscsoport. [C3, J1] *Tanulmányoztam a terheléelosztó rendszerek nem egyenletes forgalomszétosztással való továbbfejlesztésének lehetőségét azokban az esetekben, amikor a mögöttes technológia csupán egyenletes osztást tesz lehetővé az erőforrások között. Erre a célra javasoltam a virtuális erőforrás-foglalás technikát, amely képessé teszi a terheléelosztókat majdnem tetszőleges forgalomosztási arány megvalósítására. Az OSPF forgalomelvezetés eseteire bevezettem és részletesen megvizsgáltam az átfedő optimalizálás nevű módszert, amely a virtuális erőforrás-foglalásnak egy átfedő hálózatot használó konkrét megvalósítása.*

A virtuális erőforrás-foglalást és az OSPF forgalomelvezetést már röviden bemutattam az első két fejezetben. Az első téziscsoportomban megjelenő *átfedő optimalizálás* (Overlay Optimization) módszert alább definiálom.

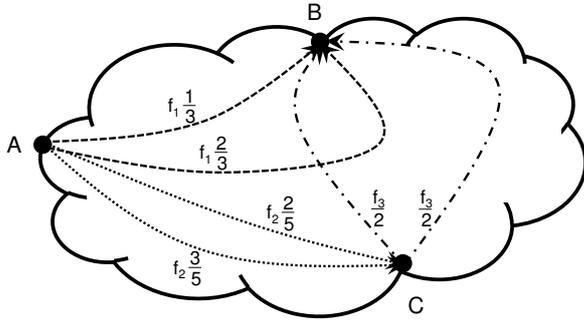
4.1.1. Definíció és tulajdonságok

1.1. tézis. *Javasoltam egy átfedő optimalizálásnak nevezett megoldást a kommunikációs hálózatokban felmerülő forgalomelvezetés problémájára, amely végponttól végpontig tartó alagutakat és ezekkel párhuzamos virtuális útvonalakat használ, és OSPF útvonalválasztást futtat ezen átfedő hálózat felett. E megoldás részeként optimalizálási feladatként formalizáltam a virtuális erőforrás-foglalás feladatát egyetlen hálózati csomópontra és egyetlen igényre. Egy példa segítségével megmutattam, hogy az OSPF forgalomelvezetés teljesítménye növelhető az átfedő optimalizálás segítségével.*

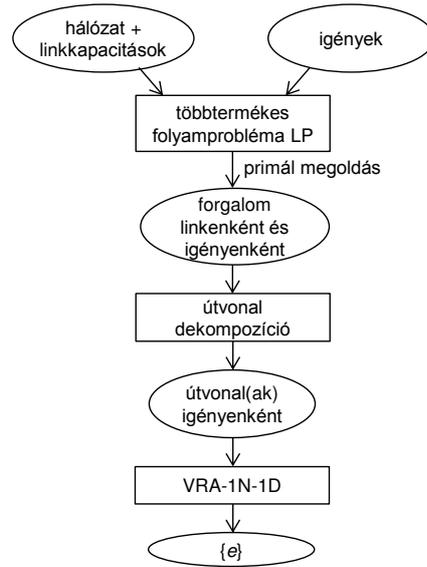
A 2(a) ábrán egy példán egy példa látható az átfedő optimalizálásra. Ebben az egyszerű tranzit hálózatban három szélső útvonalválasztó (edge router) található – A , B és C – és egy teljesen összekötött MPLS átfedő hálózat ezek felett, amely két útvonalat tartalmaz útvonalválasztó páronként. Ugyanakkor ezt az MPLS átfedő hálózatot a felette futó hagyományos OSPF útvonalválasztó protokoll egy IP hálózati topológiának látja. Tegyük fel, hogy az ideális forgalomszétosztási arányok azok, amelyek az ábrán láthatóak. Könnyen látható, hogy ez hagyományos ECMP-vel nem valósítható meg. Az általam javasolt technikával azonban kihúzhatunk négy virtuális linket (egy az $A - B$ útvonalon és hármat az $A - C$ -n), hogy pontosan a megkívánt osztásarányokat kapjuk.⁴

Az átfedő optimalizálást általánosabb esetben is vizsgálhatjuk, amikor csupán egy linkkapacitásokkal ellátott hálózat és az igények vannak megadva, és feltehetjük, hogy lehetőségünk van – akár párhuzamos – végponttól végpontig tartó alagutakat kihúzni. Ez esetben először meg kell határozni az alagutakat, majd a VRA átfedő optimalizálást kell alkalmazni ezen útvonalak felett. Az általam javasolt technika főbb lépéseit mutatja be a 2(b) ábra. Ennek legfontosabb lépése a VRA-1N-1D, amely egy csomópontra és egy igényre vonatkozó VRA-t jelent (VRA for One Node, One Demand). Jelen esetben osztás csak a forrás csomópontokban történik, ami jelentősen leegyszerűsíti a feladat analitikus megoldását, ugyanis

⁴A „virtuális útvonal” kifejezés itt helyénvalóbb lenne, de az egyszerűség kedvéért a továbbiakban is virtuális linkenek hívom ezeket.



(a) Példa átfedő optimalizálásra



(b) Átfedő optimalizálás
alagutak/útvonalak meghatározásával

2. ábra. Átfedő optimalizálás

ekkor egy VRA feladat szétbontható D darab független VRA-1N-1D feladatra, ahol D az igények száma.

Lássuk ezek után a VRA-1N-1D feladat formális definícióját! Adott egyetlen csomópont, ahol egyetlen igényhez tartozó forgalmat kell szétosztani k darab kimenő linkre (vagy útvonalra/alagútba, de az egyszerűség kedvéért a továbbiakban csak linkeként hivatkozom ezekre), linkenként rendre g_1, g_2, \dots, g_k megkívánt forgalommal (lásd az 1. táblázatot a jelölésekről). Legyen továbbá $G_0 = \sum_{i=1}^k g_i$. A cél az, hogy OSPF ECMP-t használva szétosszuk a bejövő forgalmat a kimenő linkek között oly módon, hogy a kialakuló h_1, h_2, \dots, h_k részfolyammennyiségek minél jobban megközelítsék az előírányzott g_1, g_2, \dots, g_k értékeket. A közelítés jóságának mérőszámát (a „távolságot”) linkenkénti hiba néven definiálom: $U_i = h_i/g_i$. Ezekből adódik a feladatban minimalizálandó mérőszám (U), mely a linkenkénti hibák maximuma.⁵

A cél elérése érdekében virtuális linkeket használok a fizikaiakkal párhuzamosan. Jelölje e_i a linkek számát (a fizikai és az esetleges virtuálisok együttesen) az i . fizikai link helyén, és $E = \sum_i e_i$ a kiosztott linkek összegzett darabszámát. Terjedelmi korlátok miatt csak azzal az esettel foglalkozok, amelyben nem engedjük meg egy link teljes lekapcsolását (azaz $e_i > 0$). Az OSPF ECMP egyenletesen szétosztó elvét alkalmazva azt kapjuk, hogy $h_i = G_0 e_i / \sum_{j=1}^k e_j = G_0 e_i / E$, $i = 1 \dots k$ és $U = \max_i h_i/g_i = \max_i G_0 e_i / (E g_i)$. Felteszem továbbá, hogy létezik egy felső korlát az egy igényhez kiosztott linkek összesített számára – ahogyan ez a valós útvonalválasztók esetében is van –, amelyet $E \leq Q$ -val modellezem. Ezek után a formális definíció a következő:

⁵Az U_i és U mennyiségekre hibaként hivatkozok, holott azok tulajdonképpen a valódi és a megkívánt forgalmommennyiségek hányadosai. Ez némileg szokatlan lehet, ugyanis általában a hibákat nulla közelében kívánatos tartani, ez esetben azonban $U = 1$ jelenti az ideális megoldást.

Jelölés	Magyarázat
k	a felhasznált kimenő linkek száma
$g_1, g_2, \dots, g_k \in \mathbb{Z}^+$	a forgalom megkívánt mérete kimenő linkenként
$G_0 = \sum_{i=1}^k g_i$	a forgalom teljes mérete
h_1, h_2, \dots, h_k	a valódi forgalom kimenő linkenként
$U_i = h_i/g_i$	a hiba az i . kimenő linken
$U = \max_i U_i$	egy virtuális erőforrás-foglalás hibája
e_1, e_2, \dots, e_k	a kiosztott linkek száma (fizikai és virtuális együtt)
$E = \sum_{i=1}^k e_i$	a kiosztott linkek száma összesen
$Q \in \mathbb{Z}^+$	a felső korlát a kiosztott linkek összesített számára

1. táblázat. Jelölések a VRA-1N-1D-hez

1. feladat, VRA-1N-1D. Adott k , $\{g_i\}$ és Q , keressük azt az $\{e_i\}$ -t, amelyik minimalizálja U -t úgy, hogy $\sum_i e_i \leq Q$.

Az átfedő optimalizálás hasznosságának érzékeltetésére tekintsük ismét az 1. ábrát! Az $A - B - C$ útvonallal és két $A - C$ útvonallal a maximális linkkihasználtság értéke 1,0 lesz, míg hagyományos OSPF-TE-vel 1,5-nél alacsonyabb érték nem érhető el.

1.2. tézis. *Korlátokat adtam a VRA-1N-1D feladat hibájára különböző feltételek mellett.*

A VRA-1N-1D feladat hibájára vonatkozóan bebizonyítottam a következő tételeket:

1. $U \geq 1$.
2. Ha $Q \geq G_0$ akkor $\exists \{e_i\}$ amelyre $U = 1$.
3. $U \leq G_0 / \min_i g_i$.
4. Ha G_0 nem korlátos, de E -hez létezik egy véges Q korlát, akkor U tetszőlegesen nagy lehet bármely $Q > 2$ -re.

Az első állítás egy alsó korlát. A második azt mondja ki, hogy ha a kiosztható virtuális linkek száma kellően nagy, akkor ez az alsó korlát el is érhető. A harmadik állítás egy, a teljes forgalommenyiségtől függő felső korlátot ad, a negyedik pedig egy általános felső korlát létezését cáfolja.

A bizonyítható hibakorlátok létezése önmagában is fontos, azonban e korlátoknak közvetlen gyakorlati jelentőségük is van, ugyanis a következő tézis bináris keresési algoritmusában is felhasználok ezeket.

1.1. algoritmus. VRA-1N-1D-Rögzített-E

Bemenet: $\alpha, k, \{g_i\}, E$

Kimenet: *megoldható, egyetlen_megoldás, $\{e_i\}$*

ciklus $i \leftarrow 1 \dots k$:

$$x_i \leftarrow \left\lfloor \frac{\alpha g_i E}{G_0} \right\rfloor$$

ciklus vége

ha $\sum_{i=1}^k x_i < E$ **akkor**

megoldható \leftarrow hamis

különben ha $\sum_{i=1}^k x_i = E$ **akkor**

megoldható \leftarrow igaz

egyetlen_megoldás \leftarrow igaz

különben

megoldható \leftarrow igaz

egyetlen_megoldás \leftarrow hamis

ha vége

ha *megoldható* = igaz **akkor**

Egy $\{e_i\}$ megtalálásához oldd meg a következő egyenletrendszert:

$$\sum_{i=1}^k e_i = E; \quad 1 \leq e_i \leq x_i \quad (e_i \in \mathbb{Z}, i = 1 \dots k)$$

ha vége

4.1.2. Optimális megoldás

1.3. tézis. *Adtam egy optimális, pszeudo-polinomiális futásidejű megoldást a VRA-1N-1D feladatra. Adtam továbbá egy optimális, pszeudo-polinomiális futásidejű megoldást arra a feladatváltozatra is, amelyben a linkek számát minimalizáljuk a maximális hibát adott korlát alatt tartva. Szinten optimális és pszeudo-polinomidejű algoritmusokat mutattam az előző két feladatnak azon változataira, amelyekben a hibának vagy a linkek számának a minimalizálása több csomópontban egyszerre történik úgy, hogy közös korlát érvényes rendre a linkek összesített számára vagy a hibára.*

Bebizonyítottam, hogy a lehetséges linkkiosztások száma $\binom{Q}{k}$ [Disszertáció, 7. lemma], ami azt jelenti, hogy Q kis értékeire (például $Q \leq 30 \dots 50$) járható út a kimerítő keresés, sokkal nagyobb Q -kra azonban nem. Mivel a VRA-1N-1D olyan esetekben is felhasználható, ahol a Q ezres nagyságrendben is lehet – mint például SDN-nél, ahol Q a maximális szabályszámot jelenti –, szükség van egy gyorsabb megoldásra.

Az 1.1. algoritmus [Disszertáció, 3.1. alg.] megvizsgálja adott $\alpha, k, \{g_i\}$ és E esetén, hogy lehetséges-e $U \leq \alpha$ -t teljesítve kiosztani a linkeket. Ha a kiosztás megvalósítható, akkor ad is egy megoldást, és azt is jelzi, hogy az adott megoldás az egyetlen lehetséges megoldás-e. Bebizonyítottam, hogy az 1.1. algoritmus valóban helyes megoldást ad, és hogy számítási komplexitása $O(k)$.

1.2. algoritmus. VRA-1N-1D

Bemenet: $k, \{g_i\}, Q$ **Kimenet:** $\{e_i\}, U(E), U$ $legjobb_U \leftarrow G_0 + 1.0$ **ciklus** $E \leftarrow k \dots Q$: $\{jelenlegi_e_i\}, jelenlegi_U \leftarrow \text{VRA-1N-1D-BIN-KERES}(\{g_i\}, E)$ **ha** $jelenlegi_U < legjobb_U$ **akkor** $legjobb_U \leftarrow jelenlegi_U$ $\{legjobb_e_i\} \leftarrow \{e_i\}$ **ha vége** $U(E) \leftarrow legjobb_U$ {csak a későbbi algoritmusokban használjuk}**ciklus vége** $U \leftarrow legjobb_U$ $\{e_i\} \leftarrow \{legjobb_e_i\}$

Ezután adtam egy bináris keresést alkalmazó módszert a legkisebb olyan α megtalálására, amelyre az 1.1. algoritmusnak még van megoldása adott g_i -k és E mellett [VRA-1N-1D-Bin-Keres, Disszertáció, 3.2. alg.]. Bebizonyítottam a ciklusa leállási feltételének a helyességét, valamint azt is, hogy $\log(G_0^2 E)$ lépésben fut le, ami összességében $O(k \log(G_0^2 E))$ polinomiális komplexitást jelent.

Már csak az maradt hátra, hogy megtaláljuk azt az E értéket, amelyre a hiba a legkisebb egy adott Q mellett. Erre szolgál az 1.2 algoritmus [Disszertáció, 3.3. alg.]. Ez az algoritmus elméletileg nem polinomidejű, hiszen a komplexitása $O(Qk \log(G_0^2 Q))$, ami nem polinomiális függvénye Q hosszának, $\log(Q)$ -nak. Ez a komplexitás azonban mégis kellően alacsony, így az algoritmus a gyakorlati esetekben igen jól használható.

A következő feladat a linkek számának minimalizálása adott maximális hiba mellett:

2. feladat, VRA-1N-1D-Link-Min. Adott $k, \{g_i\}$ és $U_{\text{lim}} \geq 1$, keressük azt az $\{e_i\}$ -t, amelyik minimalizálja a linkek összesített számát (E) úgy, hogy $U \leq U_{\text{lim}}$.

Tekintsük az 1.2 algoritmust $Q \leftarrow G_0$ -lal. Az ily módon készült $U(E)$ függvény (nem szigorúan) monoton csökkenő, amelynek értelmezési tartománya a pozitív egész számok egy részhalmaza. A feladat megoldása vagy az az E , amelyre $U(E - 1) > U_{\text{lim}}$ és $U(E) \leq U_{\text{lim}}$, vagy k , ha $U(k) \leq U_{\text{lim}}$. Az algoritmus komplexitása $O(G_0 k \log(G_0^3))$.

A hátralévő két feladatváltozatban több optimalizálási problémát oldunk meg egyidejűleg.

3. feladat, Párhuzamos-VRA-1N-1D. Adott k_n ($n = 1 \dots N$), $\{g_{ni}\}$ és Q , keressük azt az $\{e_{ni}\}$ -t, amelyik minimalizálja $U_{\text{max}} = \max U_n$ -t úgy, hogy $\sum_n E_n = \sum_n \sum_i e_{ni} \leq Q$.

A feladat megoldására adtam egy mohó algoritmust [Disszertáció, 3.4. alg.]. Ez a csomópontok $U(E)$ függvényeit használja, amelyeket az 1.2 algoritmus segítségével határozhatunk meg. Bebizonyítottam, hogy ez az algoritmus egy optimális megoldást ad. Meghatároztam továbbá az algoritmus komplexitását, amely $O(NQk \log(G_0^2 Q) + Q)$, ahol $k = \max k_i$, $G_0 = \max G_{0i}$.

4. feladat, *Párhuzamos-VRA-1N-1D-Link-Min*. Adott k_n ($n = 1 \dots N$), $\{g_{ni}\}$ és U_{lim} , keressük azt az $\{e_{ni}\}$ -t, amelyik minimalizálja $\sum_n E_n = \sum_n \sum_i e_{ni}$ -t úgy, hogy $U_{\text{max}} = \max U_n \leq U_{\text{lim}}$.

Ez a feladat triviálisan szétbontható N darab független VRA-1N-1D-Link-Min feladattá, amelyek a 2. feladatnál ismertetett módon oldhatóak meg.

4.2. Helyi egyenrangú optimalizálás

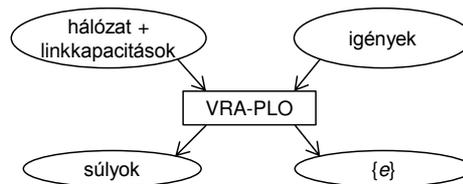
2. téziscsoport. [C1, J1] *Javasoltam és részletesen elemeztem egy másik virtuális erőforrás-foglalási módszert az OSPF forgalomelvezetés feladatára, amely nem használ átfedő hálózatot, ami megkönnyíti az alkalmazását. Ez a megközelítés az eredeti topológiát használja és a döntéseket helyi szinten, a hálózati csomópontokban hozza, ezért helyi egyenrangú optimalizálásnak neveztem el.*

A *helyi egyenrangú optimalizálás* (Peer-Local Optimization, PLO) esetében adott egy linkkapacitásokkal ellátott hálózat és egy igényhalmaz (lásd a 3. ábrát). Az optimalizálás során meghatározandó minden linkhez egy súlyérték és a párhuzamos linkek száma, úgy, hogy mindezt az OSPF-nek átadva a hálózatban a maximális linkkihasználtság a lehető legkisebb legyen. Más szavakkal tehát a helyi egyenrangú optimalizálás a VRA-val kiegészített OSPF-TE-nek [12] szükséges bemenetet állítja elő.

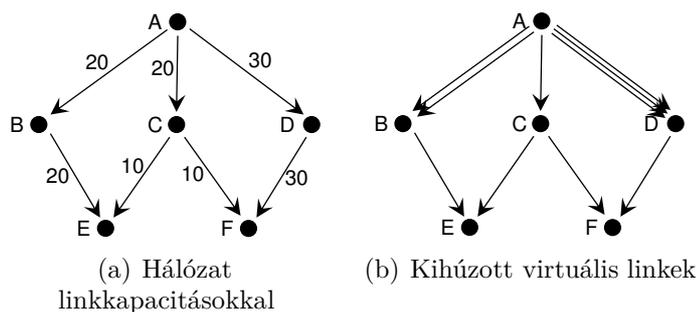
4.2.1. Definíció és tulajdonságok

Csakúgy mint eddig, ez esetben is korlátozva van a csomópontonként felhasználható linkek száma, azonban most ez a korlát *csomópontonként és igényenként* érvényes, ahogyan az útvonalválasztók esetében is az van meghatározva, hogy egyetlen adatfolyamot legfeljebb hány kimeneti linkre lehet szétosztani.

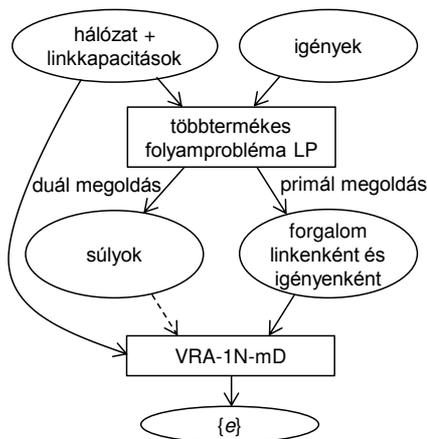
Tekintsük például a 4(a) ábrán látható hálózatot a feltüntetett linkkapacitásokkal és a következő két igénnyel: $A \rightarrow E : 30$, $A \rightarrow F : 40$. Nyilvánvalóan akkor lesz optimális az igények elvezetése, ha minden linket teljesen kihasználunk, amihez rendre $2 : 1$ és $1 : 3$ arányban szükséges szétosztani a két igényt az A csomópontban. Tételezzük fel, hogy legfeljebb $Q = 4$ darab kimenő linket használhatunk csomópontonként és igényenként. Ekkor optimális megoldáshoz jutunk, ha a virtuális linkeket a 4(b) ábrán látható módon húzzuk ki. Igaz ugyan, hogy ekkor hat link indul ki A -ból, mégis minden igényt legfeljebb négyfelé osztunk, így nem sérül a korlát.



3. ábra. Virtuális erőforrás-foglalás – helyi egyenrangú optimalizálás



4. ábra. Példa linkszámkorlátra több igény esetén. Igények: $A \rightarrow E : 30$, $A \rightarrow F : 40$



5. ábra. A helyi egyenrangú optimalizálás működése

2.1. tézis. *Javasoltam egy új megoldást az OSPF forgalomelvezetés feladatára, melyet helyi egyenrangú optimalizálásnak neveztem el. Ez a módszer nem használ átfedő hálózatot, és működése teljes egészében a hálózati csomópontokban hozott helyi döntéseken alapul. E megoldás része, hogy optimalizálási problémaként formálisan megfogalmaztam a virtuális erőforrás-foglalás feladatát egy csomópontra és több igényre.*

A helyi egyenrangú optimalizálás működésének vázlatát az 5. ábrán látható. Az első lépés ugyanaz, mint az átfedő optimalizálás esetében: egy többtermékes folyamproblémát leíró lineáris program megoldása szétosztható folyamokkal, amelyről tudjuk, hogy polinom időben elvégezhető. A primál megoldás megadja linkenként és igényenként a forgalom méretét, a duál megoldásból pedig meghatározhatóak az OSPF-TE-hez szükséges linksúlyok. Végül az utolsó lépés a VRA-1N-mD feladat megoldása minden csomópontra külön-külön. A VRA-1N-mD rövidítés jelentése virtuális erőforrás-foglalás egy csomópont és több igény esetén (Virtual Resource Allocation for One Node and multiple Demands), és lokálisan optimális virtuális linkkonfigurációt eredményez.

Következzék most a VRA-1N-mD feladat formális definíciója, melyhez a jelöléseket a 2. táblázatban foglaltam össze.

Jelölés	Magyarázat
k	a felhasznált kimenő linkek száma
D	az igények száma
$G = (g_{ij}) \in \mathbb{Z}^{D \times k}$	a forgalom megkívánt mérete igényenként és kimenő linkenként ($g_{ij} \geq 0$)
$\Gamma = (\gamma_{ij}) \in \mathbb{R}^{D \times k}$	a G mátrix soronként normalizált változata
$\Sigma = (\sigma_{ij}) \in \{0, 1\}^{D \times k}$	$\sigma_{ij} = 0$ ha $g_{ij} = 0$, $\sigma_{ij} = 1$ egyébként
$G_i = \sum_{j=1}^k g_{ij}$	az i igény teljes forgalma
h_{ij}	a valódi forgalom igényenként és kimenő linkenként
e_1, e_2, \dots, e_k	a kiosztott linkek száma (fizikai és virtuális együtt)
$E_i = \sum_{j=1}^k e_j \sigma_{ij}$	az i igényhez a legrövidebb utakon lévő párhuzamos linkek összesített száma
$U_{ij} = e_j / (\gamma_{ij} E_i)$	a hiba igényenként és linkenként (csak ahol $\gamma_{ij} > 0$)
$U = \max_{\gamma_{ij} > 0} U_{ij}$	a csomópontonkénti hiba
$Q \geq E_i \ (\forall i), \ Q \in \mathbb{Z}^+$	a felső korlát az igényenként felhasználható linkek számára

2. táblázat. Jelölések a VRA-1N-mD-hez

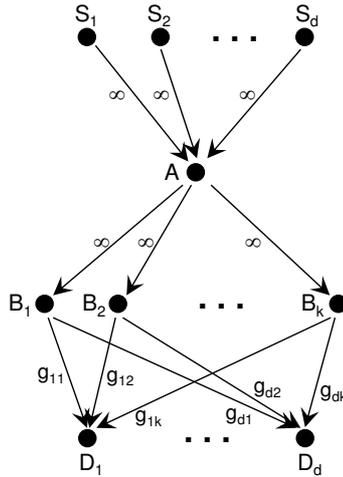
Egy A hálózati csomóponthoz adott a

$$G = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1k} \\ g_{21} & g_{22} & \dots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{D1} & g_{D2} & \dots & g_{Dk} \end{bmatrix}, \quad (1)$$

mátrix, amely megadja, hogy hogyan kell szétosztani a csomóponthoz bejövő $1 \dots D$ igényeket az $1 \dots k$ kimenő linkek között: g_{ij} az a forgalommenyiség, amely az i igényhez tartozik és a j linken kell kiküldeni ($g_{ij} \in \mathbb{Z}$, $g_{ij} \geq 0$). Az egyszerűség kedvéért feltehetjük, hogy G nem tartalmaz csupa nullákból álló sort vagy oszlopot. A későbbiekben G -nek a soronként normalizált és a szignum változatait is használni fogom:

$$\gamma_{ij} = \frac{g_{ij}}{\sum_{n=1}^k g_{in}}; \quad \sigma_{ij} = \begin{cases} 0, & \text{ha } g_{ij} = 0 \\ 1, & \text{ha } g_{ij} > 0 \end{cases}.$$

Az A csomópont j linkjéhez rendelt párhuzamos linkek száma e_j , amely tartalmazza a fizikai és a virtuális linkeket is. Felteszem, hogy egy meglévő linket nem lehet letiltani, azaz $e_j > 0$. Legyen $E_i = \sum_{j=1}^k e_j \sigma_{ij}$ az i igényhez tartozó legrövidebb utakon lévő párhuzamos linkek összesített száma és $G_i = \sum_{j=1}^k g_{ij}$ az igényhez tartozó felajánlott forgalom. Az ECMP egyenletes szétosztási szabálya miatt az igényenkénti forgalommenyiség egy kimenő linken $h_{ij} = e_j G_i / E_i$. Az igényenkénti és linkenkénti hiba (U_{ij}) legyen az i igényhez tartozó és a j kimenő linken átvitt forgalommenyiség (h_{ij}) elosztva a megfelelő felajánlott forgalommenyiséggel (g_{ij}), amely azonban csak akkor definiált, ha a felajánlott forgalom nem nulla:



6. ábra. Egy adott G mátrixhoz tartozó hálózat, linkkapacitásokkal

$U_{ij} = h_{ij}/g_{ij} = e_j/(\gamma_{ij}E_i)$, $\forall g_{ij} > 0$. A csomópontonkénti hibát – vagy röviden csak hibát – az igényenkénti és linkenkénti hibák maximumaként definiálom: $U = \max_{i,j:\gamma_{ij}>0} U_{ij}$.

Összefoglalva, a feladatot a következőképp formalizálhatjuk:

5. feladat, *VRA-1N-mD*. Adott k , D , G és Q , keressük azt az $\{e_j\}$ -t, amelyik minimalizálja U -t úgy, hogy $E_i \leq Q$ minden $i = 1 \dots D$ -re.

Első közelítésben azonban egy egyszerűbb változatát vizsgálom a problémának:

6. feladat, *VRA-1N-mD-Korlátlan*. Adott k , D és G , keressük azt az $\{e_j\}$ -t, amelyik minimalizálja U -t.

Vegyük észre, hogy ez utóbbi esetben a feladatot pusztán a G mátrixszal adjuk meg.

2.2. tézis. *Megmutattam, hogy a G mátrix, amely a VRA-1N-mD és a VRA-1N-mD-Korlátlan feladatok gerincét alkotja, majdnem tetszőleges lehet: bármely nemnegatív mátrix, melynek van legalább egy nullától különböző eleme minden sorában és oszlopában, G mátrixa lehet egy adott csomópontnak egy alkalmasan választott hálózatban.*

Ez a tézis a disszertációm 11. tételén alapszik. A bizonyításban egy adott G mátrixhoz egy hálózatot készíték, melyben a helyi egyenrangú optimalizálás után (ld. az 5. ábrát) egy adott csomópontban a szükséges osztásarányok éppen a G mátrixban adottakkal egyeznek meg. Legyen G az (1) képlet alakjában adott. Az ehhez tartozó kapacitásos hálózat a 6. ábrán látható. Összesen d igény van a rendszerben, melyek közül az i -edik S_i -ből D_i -be megy és az erre az igényre vonatkozó megkívánt osztásarányt g_{ij} adja meg ($j = 1 \dots k$).

Azért is fontos ez az eredmény, mert lehetővé teszi, hogy ezután pusztán a mátrixokra koncentráljunk az esetleg jóval bonyolultabb hálózatok helyett.

2.3. tézis. *Korlátokat adtam a VRA-1N-mD és a VRA-1N-mD-Korlátlan feladatok hibájára, valamint mutattam egy polinomiális idejű algoritmust, amely eldönti, hogy korlátlan számú linket felhasználva az általános alsó korlát elérhető-e az adott konkrét feladatra.*

A VRA-1N-mD és a VRA-1N-mD-Korlátlan feladatok hibájával kapcsolatban a következő állításokat bizonyítottam be [Disszertáció, 12–14. lemmák]:

1. $U \geq 1$.
2. $U \leq (\min_{i,j:\gamma_{ij}>0} \gamma_{ij})^{-1}$.
3. Nem létezik általános (G -től független) felső korlát a csomópontonkénti hibára.

Ezek a kijelentések tehát igazak mind a korlátos, mind pedig a korlátlan számú linket használó feladatváltozatra.

Egy G mátrixot *konzisztensnek* nevezek akkor és csak akkor, ha $U = 1$ elérhető egy megfelelő $\{e_j\}$ -vel, megengedve korlátlanul sok párhuzamos link felhasználását. Mutattam egy algoritmust, amely eldönti, hogy egy adott G konzisztens-e, és ha az, akkor ad egy $\{e_j\}$ -t, amelyre $U = 1$ [Disszertáció, 4.1. alg.]. Ezen algoritmus komplexitása $O(d^2k)$, ami azt jelenti, hogy polinomidőben fut le.

4.2.2. Korlátlan számú párhuzamos link

Tekintsük most VRA-1N-mD-Korlátlan feladatot, azaz a VRA-1N-mD megoldását keressük korlátlan számú párhuzamos linkkel. Az egyszerűség kedvéért ebben az alfejezetben a linkek számának (e_j -nek) a normalizált változatát használom, amelyet a félreértések elkerülése végett f_j -vel jelölök: $f_j = e_j / \sum_{i=1}^k e_i$ ($f_j \in \mathbb{R}^+$, $\sum_{j=1}^k f_j = 1$).

2.4. tézis. *Bebizonyítottam, hogy a VRA-1N-mD-Korlátlan feladatra véges számú link felhasználásával nem adható minden esetben optimális linkkiosztás.*

Valójában azt mutattam meg, hogy létezik legalább egy VRA-1N-mD-Korlátlan feladat, amelynek a G mátrixa egész számokból áll, mégis az egyetlen optimális megoldásban az f_j -k irracionális számok [Disszertáció, 16. tétel]. Az állítás bizonyításában megmutatom, hogy a

$$G = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

mátrixszal megadott feladatnak egyetlen optimális megoldása a következő:

$$f_1 = \frac{2}{5}(7 - \sqrt{34}), f_2 = \frac{1}{5}(-16 + 3\sqrt{34}), f_3 = \frac{1}{5}(7 + \sqrt{34}),$$

ami bizonyítja e tézis állítását is.

2.5. tézis. *Bebizonyítottam, hogy egyetlen algoritmus sem adhat optimális megoldást a VRA-1N-mD-Korlátlan feladatra véges sok lépésben, még akkor sem, ha a lépésszám függhet a konkrét feladattól.*

Egy megoldás f_1, f_2, \dots, f_k alakban áll elő, ahol $f_j \in \mathbb{R}^+$. Természetes igény, hogy egy algoritmus a valós f_j konstansokat valamilyen zárt formában adja meg, azonban a „zárt forma” fogalma sokféleképpen definiálható. A továbbiakban ezért azt várom el, hogy az

f_j -k véges hosszú kifejezésekkel legyenek megadva, amelyekben csak egész konstansok és a szokásos $+$, $-$, \cdot , $/$ és az n -edik gyökvonás ($n \in \mathbb{Z}^+$) operátorok szerepelnek.

Ez a tézis a disszertációm 18. tételén alapul, amely azt állítja, hogy létezik legalább egy VRA-1N-mD-Korlátlan feladat, amelynek az egyetlen optimális megoldásában szerepel legalább egy f_j , amely nem írható fel a fenti módon definiált zárt alakban.

Az alapgondolat tehát az, hogy az optimális megoldás nem számítható ki véges sok lépésben, ha le sem írható zárt alakban, hiszen az eredmény leírása is része a megoldásnak. A tétel bizonyításában a következő mátrixot vizsgáltam:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 & 0 & 0 \\ 6 & 6 & 1 & 0 & 0 & 0 \\ 6 & 6 & 6 & 1 & 0 & 0 \\ 6 & 6 & 6 & 6 & 1 & 0 \\ 6 & 6 & 6 & 6 & 6 & 1 \end{bmatrix} .$$

Bebizonyítottam, hogy e mátrixra f_1 a következő polinomegyenlet megoldásaként áll elő:

$$923\,521f_1^5 - 16\,980\,870f_1^4 + 118\,664\,280f_1^3 - 390\,577\,680f_1^2 + 934\,673\,904f_1 - 336\,117\,600 = 0 . \quad (2)$$

A Maple [18] matematikai szoftver segítségével megmutattam, hogy ennek a polinomnak egyetlen valós gyöke van (és négy komplex). A Galois elmélet [19] szerint egy polinomegyenlet megoldásai akkor és csak akkor írhatóak fel gyökképlettel⁶, ha a polinom Galois csoportja feloldható. A Maple-t használva meghatároztam, hogy a (2) egyenletben szereplő polinom Galois csoportja az S_5 szimmetrikus csoport. Ez a 120 elemű csoport nem feloldható, ami azt jelenti, hogy (2) megoldásai nem írhatóak fel gyökképlettel, ami pedig bizonyítja az állításunkat.

2.6. tézis. *Adtam egy közelítő algoritmust, amely a VRA-1N-mD-Korlátlan feladat optimális megoldásához tetszőlegesen közeli megoldást talál polinomidőben.*

Első közelítésben felírtam a 2.1. lineáris programot (továbbiakban: LP) [Disszertáció, 4.2. LP], amely kiszámolja az $\{f_j\}$ -t, úgy, hogy a csomópontonkénti hibát egy adott α konstans alatt tartja.⁷ Természetesen túl kicsi α -kra az LP-nek nem lesz megoldása.

Következő lépésként bináris keresést javasoltam annak a legkisebb α -nak a megtalálására, amelyre a 2.1. LP még megoldható [Disszertáció, 4.2. alg.]. Ennek az algoritmusnak van egy bemeneti paramétere, ϵ_U , amely azt adja meg, hogy milyen közel akarunk jutni az optimális

⁶Azaz a megoldások véges alakban felírhatóak pusztán egész konstansokkal és az $+$, $-$, \cdot , $/$ valamint az n -edik gyökvonás ($n \in \mathbb{Z}^+$) operátorok segítségével.

⁷Ebben az LP-ben ideiglenesen feloldottam a $\sum f_j = 1$ korlátozást, és az egyértelműség kedvéért bevezettem az \hat{f}_j változókat. Mindezt azért tettem, hogy az $f_j > 0$ -t kikényszerítsem – ami azt jelenti, hogy egy link nem letiltható – és ennek ez volt a legegyszerűbb módja.

LP 2.1 VRA-1N-mD-Korlátlan, adott α

$$\begin{aligned} \text{indexek: } & i = 1 \dots D \\ & j = 1 \dots k \\ \text{konstansok: } & \alpha \quad (\alpha \geq 1, \alpha \in \mathbb{R}) \\ & \gamma_{ij} \quad (\gamma_{ij} \in \mathbb{Q}, \gamma_{ij} \geq 0, \forall i : \sum_{n=1}^k \gamma_{in} = 1) \\ & \sigma_{ij} = \text{sgn}(\gamma_{ij}) \\ \text{változók: } & \hat{f}_j \quad (\hat{f}_j \geq 1, \hat{f}_j \in \mathbb{R}) \\ \text{célfüggvény: } & \min. \sum_{j=1}^k \hat{f}_j \\ \text{feltételek: } & 0 \leq \sum_{n=1}^k \sigma_{in} \hat{f}_n - \frac{\hat{f}_j}{\gamma_{ij} \alpha}, \quad \forall i, j : \gamma_{ij} > 0 \end{aligned}$$

hibához. A 2.4. tézisből következik, hogy ennek megadása feltétlenül szükséges, ugyanakkor így valóban tetszőlegesen meg tudjuk közelíteni az optimális megoldást.

Bebizonyítottam, hogy e módszer számítási komplexitása polinomiális. Az alkalmazott lineáris program ugyanis nem tartalmaz egész értékű változókat, így polinomidőben megoldható. Ezt az LP-t $\log_2(1/(\epsilon_U \min_{i,j} \gamma_{ij}))$ -szer kell megoldani, ami azt jelenti, hogy a teljes módszer valóban polinomiális.

E tézisben tehát adtam egy olyan iteratív megoldást a VRA-1N-mD-Korlátlan feladatra, amely gyorsan konvergál az optimumhoz. Ez jelentős eredmény, hiszen a 2.5. tézis kimondja, hogy ennél lényegesen jobb megoldás nem létezhet.

4.2.3. Korlátos számú párhuzamos link

2.7. tézis. *Adtam a VRA-1N-mD feladatra két különböző, egész értékű lineáris programon alapuló optimális megoldást. Adtam továbbá egy pszeudopolinomiális futásidejű heurisztikát ugyanerre a problémára.*

Mivel a VRA-1N-mD feladat megoldásához felhasználható párhuzamos linkek száma korlátos, ezért mindig van legalább egy optimális megoldás – hiszen csak véges számú linkkiosztás létezik, amelyek között mindig van legalább egy legjobb. Nyilván minden linkkiosztás megengedett, amelyre $e_j > 0 \forall j$ és $\sum_{j=1}^k e_j \leq Q$. Létezhetnek továbbá olyan megengedett linkkiosztások is – ahogy a 4.2.1 alfejezet elején megmutattam – amelyekre $e_j > 0 \forall j$ mellett $E_i \leq Q \forall i$ teljesül, de $\sum_j e_j \leq Q$ nem teljesül. Ez azt jelenti, hogy legalább $\binom{Q}{k}$ megengedett linkkiosztás létezik – ennek magyarázatát lásd az 1.3. tézisnél –, szükség van tehát az egyszerű kimerítő keresésnél egy hatékonyabb megoldásra.

Adtam ezért egy egész értékű lineáris programot⁸, amely optimális megoldást talál a VRA-1N-mD feladatra [Disszertáció, 4.4. LP]. Ez az ILP azonban nagyon sok egész értékű

⁸Az egész értékű lineáris programokat az Integer Linear Program rövidítésével továbbiakban ILP-knek nevezem.

segédváltozót használ, ami megnöveli a futásidőt, ezért javasoltam egy gyorsabb megoldást is.

Ez a második módszer egy ILP-alapú iteratív megoldás. Először is a korlátos linkszámú feladatváltozatnak megfelelően módosítottam a 2.1. LP-t: hozzáadtam a Q konstanst, a valós értékű f_j változókat a pozitív egész értékű e_j -kre cseréltem, végül pedig egy új feltételt is felvettem: $\sum_{j=1}^k \sigma_{ij} e_j \leq Q$ ($\forall i$) [Disszertáció, 4.3. LP]. Ezt az ILP-t egy bináris keresésbe ágyazottan ismételtlen alkalmazva a VRA-1N-mD feladat tetszőlegesen jó közelítését kapjuk [Disszertáció, 4.4. alg.]. Ez esetben azonban a lehetséges linkkiosztások száma véges, így lehetőség van két linkkiosztás hibájának a különbségére abszolút alsó korlátot adni: bebizonyítottam, hogy $\Delta U \geq (Q \max_{i,j} g_{ij})^{-2}$ [Disszertáció, 21. lemma]. Ezt a konstanst a bináris keresés leállási feltételében alkalmazva egy *optimális megoldást* fogunk találni.

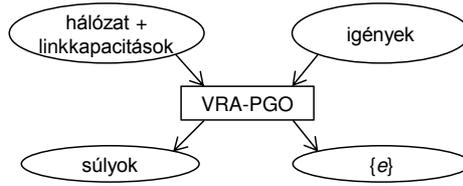
Ez utóbbi iteratív megoldás a gyakorlatban sokkal gyorsabbnak bizonyult az első, monolitikus ILP-nél. Ugyanakkor e második módszer is egy ILP-n alapszik, így a polinomiális futási idő nem garantálható. Éppen ezért javasoltam egy harmadik, heurisztikus algoritmust is [Disszertáció, 4.5. alg.]. Ennek alapötlete az, hogy a VRA-1N-mD feladat G mátrixát valahogy egy k hosszú vektorral reprezentáljuk, és azt g bemeneti vektorként használva oldjuk meg a VRA-1N-1D feladatot. Az előző alfejezetben leírtak szerint ez utóbbi gyorsan megoldható, így már csak az maradt hátra, hogy találjunk egy hatékony leképezést a G mátrixról a g vektorra. A VRA-1N-mD-Korlátlan feladat megoldása pontosan ezt biztosítja: annak kimeneti f_j -it tekinthetjük egyszerűen a g vektornak. Természetesen a mátrixról vektorra alakítás információvesztéssel jár, ami szuboptimális eredményhez vezethet. Ez azonban megengedhető, hiszen ez a módszer csupán egy heurisztika.

4.3. Globális egyenrangú optimalizálás

A helyi egyenrangú optimalizálásnak van egy alapvető hiányossága. A problémát az okozza, hogy a lokális optimalizálások hibái továbbterjednek az adatfolyamokkal, amelyek újabb tökéletlen osztópontokon haladnak keresztül, így a helyi hibák gyengíthetők, de erősíthetők is egymás hatására. Az effajta kaszkádhibák nem kívánt hatása elkerülhető, ha a helyi hibákat központosítva, egyidejűleg minimalizáljuk. Ezt nevezem *globális egyenrangú optimalizálásnak* (Peer-Global Optimization, PGO).

3. téziscsoport. [J1] *Javasoltam és részletesen megvizsgáltam egy másik megoldást az OSPF forgalomelvezetés feladatára, amelyet globális egyenrangú optimalizálásnak neveztem el. Bebizonyítottam, hogy a probléma NP-teljes és hatékony közelítő algoritmus sem adható rá. Adtam továbbá egy egész értékű lineáris programot, amely talál egy optimális megoldást, és rámutattam, hogy a helyi egyenrangú optimalizálás felhasználható ez esetben mint gyorsabb heurisztika.*

Az itt vizsgált optimalizálási feladat megegyezik a helyi egyenrangú optimalizálásával: adva van egy linkkapacitásokkal ellátott hálózat és egy igényhalmaz, meghatározandó minden linkhez egy súly és a párhuzamos linkek száma úgy, hogy mindezek együtt minimalizálják a maximális linkkihasználtságot (7. ábra). Az itteni megközelítésben azonban a feladatot az egész hálózatra egyidejűleg oldjuk meg, hogy így eljuthassunk az elméleti legjobb virtuális erőforrás-kiosztáshoz.



7. ábra. Virtuális erőforrás-foglalás – globális egyenrangú optimalizálás

Jelölés	Magyarázat
V	a hálózati csúcsok (csomópontok) halmaza
F	a hálózati élek (fizikai linkek) halmaza
S_n	az $n \in V$ csomópontból kiinduló (fizikai) linkek halmaza
$ S_n $	az n csomópontból kiinduló (fizikai) linkek száma
$c_l \in \mathbb{Q}^+$	az $l \in F$ link kapacitása
$w_l \in \mathbb{Q}^+$	az l link súlya
$h_l \geq 0$ ($h_l \in \mathbb{Q}$)	az l linken megjelenő teljes valódi forgalom mérete
$e_l > 0$ ($e_l \in \mathbb{Z}$)	az l link helyén lévő párhuzamos linkek száma (fizikai és virtuálisak együttesen)
$E_n = \sum_{l \in S_n} e_l$	az n csomópontból kiinduló (fizikai és virtuális) linkek száma
D	az igények száma
$O_d \in F$	a d ($1 \leq d \leq D$) igény forráscsomópontja
$D_d \in F$	a d igény célsomópontja
$G_d \in \mathbb{Q}^+$	a d igényhez tartozó forgalom mennyiség
$R \geq 0$ ($R \in \mathbb{Z}$)	a virtuális linkek csomópontonkénti maximális száma
$\beta \in \mathbb{Q}^+$	a maximális linkkihasználtság

3. táblázat. Jelölések a VRA-PGO-hoz

4.3.1. Definíció

3.1. tézis. *Azonosítottam és formálisan leírtam a globális egyenrangú optimalizálás feladatot, amely közel optimális megoldást tud nyújtani az OSPF forgalomelvezetés problémájára.*

A formális definíció előtt nézzük meg a globális egyenrangú optimalizáláshoz használt jelöléseket, melyeket a 3. táblázatban foglaltam össze. Adott egy *hálózatot* reprezentáló (V, F) irányított gráf c_l *linkkapacitással* minden l linkhez, továbbá adott egy *igényhalmaz*, minden igényhez egy forrás- és célsomóponttal, valamint a felajánlott forgalom mennyiséggel: $\{O_d, D_d, G_d\}_{d=1}^D$. Adva van még az egy csomópontban felhasználható *virtuális linkek maximális száma* (R). Keressük minden l linkhez a w_l *linksúlyt* és az e_l *párhuzamos linkek számát* (beleértve a fizikai és a virtuális linkeket is), amelyek együtt minimalizálják a $\beta = \max_{l \in F} h_l / c_l$ *maximális linkkihasználtságot*, betartva az $E_n \leq |S_n| + R$ ($\forall n \in V$) feltételt.

Az előző két alfejezetben célszerűbb volt a kiosztott linkek összesített számát (beleértve a fizikaiakat és virtuálisakat is) korlátozni, megkövetelve, hogy $E \leq Q$ és $E_i \leq Q$. A numerikus vizsgálat során azonban célravezetőbb a virtuális linkek számát limitálni, hiszen

a különböző csomópontok különböző számú kimenő fizikai linkkel rendelkezhetnek. Emiatt ebben az alfejezetben általában a virtuális linkek száma a korlátos ($E_n - |S_n| \leq R$) és külön jelölöm ahol ettől eltérnek. Megjegyzendő, hogy az egyszerű $R = Q - k$ helyettesítéssel a 4.1. alfejezetben is lehetőség van a virtuális linkek számát korlátozni ($E - k \leq R$). A 4.2. alfejezet legtöbb állítása a VRA-1N-mD feladat korlátlan linkszámú változatáról szól, de a többit az előzőhöz hasonlóan triviális átfogalmazni arra az esetre, amikor a virtuális linkek száma a limitált.

A globális egyenrangú optimalizálás feladatát ezek után a következőképpen formalizálhatjuk:

7. feladat, VRA-PGO. Adott (V, F) , $\{c_l\}$, D , $\{O_d, D_d, G_d\}$ és R , keressük azt a $\{w_l\}$ -t és $\{e_l\}$ -t, amelyek minimalizálják β -t úgy, hogy $E_n \leq |S_n| + R$ ($\forall n \in V$).

4.3.2. Optimális megoldás

3.2. tézis. *Adtam egy egész értékű lineáris programot, amely egy optimális megoldást talál a globális egyenrangú optimalizálás feladatára.*

Ezt a lineáris program, melyet a disszertációmban LP 5.1 sorszám alatt mutattam be, meglehetősen terjedelmes: 13 sor konstans, 8 sor változót és 15 sor feltételt tartalmaz, ahol egy sor jelenthet egy darabot, de egy nagyobb halmazt is. Ajánlásokat tettem továbbá azon konstansok értékeire, amelyeket ebben az ILP-ben vezettem be, úgy, hogy azok a gyakorlati számítások során jól használhatóak legyenek.

Ennek az ILP-nek az a jelentősége, hogy elméleti alsó korlátot ad a hibára. A gyakorlatban ugyanakkor nagyobb hálózatok esetén csak igen lassan jutunk el a megoldáshoz, mivel sok egész értékű segédváltozót használ. Szerencsére mégis meg tudtam oldani sok gyakorlati példahálózatra, és így módon a heurisztikák teljesítményelemzéséhez szükséges referenciaértékekhez jutottam (ld. majd a 4.3.4 alfejezetben).

E tézisben adtam egy lassú, de optimális megoldást a globális egyenrangú optimalizálás feladatára. Vegyük észre, hogy a helyi egyenrangú optimalizálásra mutatott algoritmusok felfoghatók gyorsabb, de szuboptimális heurisztikaként ugyanerre a globális feladatára. A következő alfejezetben azt mutatom meg, hogy egy gyors és akár csak közel optimális megoldást keresni reménytelen vállalkozás a feladat eredendő bonyolultsága miatt.

4.3.3. Számítási komplexitás

3.3. tézis. *Bebizonyítottam a globális egyenrangú optimalizálás feladatnak és több változatának az NP-teljességét.*

Az NP-teljességi bizonyításhoz először kissé átfogalmaztam a globális egyenrangú optimalizálás feladatát és eldöntési problémaként mondtam ki azt. Ezen felül ebben az első közelítésben a linksúlyokat bemeneti paramétereknek tekintettem:

8. feladat, Virtuális erőforrás-foglalás – globális egyenrangú optimalizálás adott élsúlyokkal (VRA-PGO-GW)⁹.

⁹Virtual Resource Allocation–Peer-Global Optimization with Given Weights

PÉLDÁNY. Egy (V, F) irányított gráf, amely egy *hálózatot* reprezentál $c_l \in \mathbb{Q}^+$ kapacitásokkal és $w_l \in \mathbb{Q}^+$ súlyokkal minden $l \in F$ linkhez. Egy $\{O_d \in F, D_d \in F, G_d \in \mathbb{Q}^+\}_{d=1}^D$ igényhalmaz. Az egy csomópontban kiosztható *virtuális linkek maximális száma*: $R \in \mathbb{Z}$ ($R \geq 0$). A *maximális linkkihasználtság*: $\beta \in \mathbb{Q}^+$.

KÉRDÉS. Létezik-e olyan linkkiosztás, amelyben $e_l > 0$ ($e_l \in \mathbb{Z}$) darab linket rendelünk minden $l \in F$ fizikai linkhez úgy, hogy $E_n \leq |S_n| + R$ ($\forall n \in V$) és $\max_l h_l/c_l \leq \beta$?

A fenti KÉRDÉS-ben $|S_n|$ kiszámítható (V, F) -ből; E_n és h_l kiszámítható (V, F) -ből, $\{e_l\}_{l \in F}$ -ből, $\{w_l\}_{l \in F}$ -ből és az igényhalmazból. Az egyetlen nem triviális rész a h_l kiszámítása, de ez is megtehető polinomidőben a disszertációmban Alg. 5.1. sorszámmal bemutatott algoritmus segítségével.

Ezen a definíción sokféleképpen lehet változtatni, hogy ezáltal a feladat különböző variánsaihoz jussunk. Néhány ilyen lehetőséget felsorolok alább:

- VRA-PGO: Ebben az esetben a linksúlyok meghatározása is része a feladatnak és ezáltal közvetve az igények elvezetéséhez szükséges útvonalválasztás is. Ez a változat hasonló a VRA-PGO-GW-hez, csak a linksúlyok kerülnek át a PÉLDÁNY-ból a KÉRDÉS-be.
- VRA-PGO-GW-SD (Single Demand, egyetlen igény): Ebben a változatban csak egyetlen forrás-cél-forgalom mennyiség hármass szerepel ($D = 1$).
- VRA-PGO-GW-Q: A VRA-PGO-GW definíciója tartalmaz egy korlátot a virtuális linkek számára ($E_n - |S_n| \leq R$). E változatban ehelyett a fizikai és virtuális linkek együttes száma van limitálva ($E_n \leq Q$).
- VRA-PGO-GW-ABS (Absolute Error, abszolút hiba): Ez esetben a β relatív hiba (kihasználtság) helyett a δ abszolút hibát használjuk, megkövetelve, hogy $\max_l (h_l - c_l) \leq \delta$.

A fenti definíciók kombinálásával a VRA-PGO feladatnak sok további, szintén jól használható változata állítható elő. Szerencsére a számítási komplexitással kapcsolatos bizonyításaim viszonylag könnyen kiterjeszthetők ezeknek az eseteknek a jelentős részére.

Bebizonyítottam, hogy a VRA-PGO-GW feladat NP-teljes [Disszertáció, 22. tétel]. A bizonyításban először beláttam, hogy a feladat NP-beli, majd azt, hogy NP-nehéz. Ez utóbbi, bonyolultabb részben Karp-redukcióval vezettem vissza a 3SAT feladatot a VRA-PGO-GW-re. Ennek módosításaival bebizonyítottam a VRA-PGO-GW-Q és a VRA-PGO-GW-ABS feladatok NP-teljességét is. Disszertációm C.3.1. függelékében mutattam a [22. tétel]-hez egy másik bizonyítást, amelyet módosítva beláttam a VRA-PGO feladat NP-teljességét is.

Végül pedig bebizonyítottam, hogy a VRA-PGO-GW-SD feladat is NP-teljes [26. tétel]. Ez az állítás nagyon fontos, ugyanis erre épül majd a következő két tézis. Maga a bizonyítás a [22. tétel] első bizonyításának a kibővítése, így szintén a 3SAT-nak a VRA-PGO-GW-SD-re való visszavezetésén alapszik.

3.4. tézis. *Megfogalmaztam a globális egyenrangú optimalizálás két változatát NP optimalizálási problémaként. Bebizonyítottam, hogy ezek optimális megoldásait nem lehetséges számítási komplexitás szempontjából hatékonyan közelíteni minden konstans szorzó erejéig (feltéve, hogy $P \neq NP$).*

Ahhoz, hogy egy feladat optimális megoldásának a közelíthetőségét bonyolultságelméleti szempontból vizsgálni lehessen, első lépésben a feladatot NP optimalizálási (NPO, [20]) feladatként kell megfogalmazni:

9. feladat, Minimális hibájú virtuális erőforrás-foglalás – globális egyenrangú optimalizálás adott élsúlyokkal (MIN-VRA-PGO-GW, röviden MVPG)¹⁰.

PÉLDÁNY. Egy (V, F) irányított gráf, amely egy *hálózatot* reprezentál $c_l \in \mathbb{Q}^+$ kapacitásokkal és $w_l \in \mathbb{Q}^+$ súlyokkal minden $l \in F$ linkhez. Egy $\{O_d \in F, D_d \in F, G_d \in \mathbb{Q}^+\}_{d=1}^D$ igényhalmaz. Az egy csomópontban kiosztható *virtuális linkek maximális száma*: $R \in \mathbb{Z}^+$.

MEGOLDÁS. Egy virtuális erőforrás-foglalás, amelyben $e_l > 0$ ($e_l \in \mathbb{Z}$) darab linket rendelünk minden $l \in F$ fizikai linkhez úgy, hogy $E_n \leq |S_n| + R$ ($\forall n \in V$).

MÉRŐSZÁM. A *maximális linkkihasználtság*: $\beta = \max_l h_l/c_l$.

CÉL. A mérőszám minimalizálása.

10. feladat, Minimális hibájú VRA-PGO adott élsúlyokkal, egyetlen igényre (MIN-VRA-PGO-GW-SD, röviden MVPGS)¹¹.

Ez a feladat megegyezik az MVPG-vel, kivéve, hogy itt pontosan egy igény van csak.

Először bemutattam, hogy mindkét fenti feladat megfelel az NPO követelményeinek [20]. Ezután bebizonyítottam, hogy nem létezik olyan polinomidejű algoritmus, amely az MVPG optimumát $6/5$ -ös szorzónál jobban közelíti [Disszertáció, 27. tétel]. Ennek bizonyítása a VRA-PGO-GW NP-teljességének [22. tétel] az első bizonyításának a kibővítése. Hasonlóképpen megmutattam, hogy nem létezik olyan algoritmus, amely az MVPGS optimumát polinomidőben $18/17$ -es szorzónál jobban közelíti [28. tétel]. A bizonyítás a VRA-PGO-GW-SD NP-teljességének [26. tétel] a bizonyításán és az MVPG közelíthetlenségének [27. tétel] a bizonyításán alapul.

A fentiekből következik, hogy nem lehetséges e két feladatnak az optimális megoldásait számítási komplexitás szempontjából hatékony algoritmussal közelíteni *minden* konstans szorzó erejéig (feltéve persze, hogy $P \neq NP$). Más szóval ez azt jelenti, hogy *az MVPG és az MVPGS feladatok nem részei a polinomidejű approximációs séma (PTAS)¹² osztálynak.*

3.5. tézis. *Bebizonyítottam, hogy az MVPGS feladatnak (amely a globális egyenrangú optimalizálás egy változata) az optimális megoldása nem közelíthető polinomidejű algoritmussal semmilyen konstans szorzó erejéig (feltéve, hogy $P \neq NP$).*

Ez a tézis a disszertációm 29. tételén alapszik. A bizonyítás a „megközelíthetlenségi rés felnagyításának technikáját” alkalmazza, amelyet nemrég javasoltak az OSPF ECMP linksúlykiosztás feladatának egy hasonló megközelíthetlenségi bizonyítása kapcsán [13].

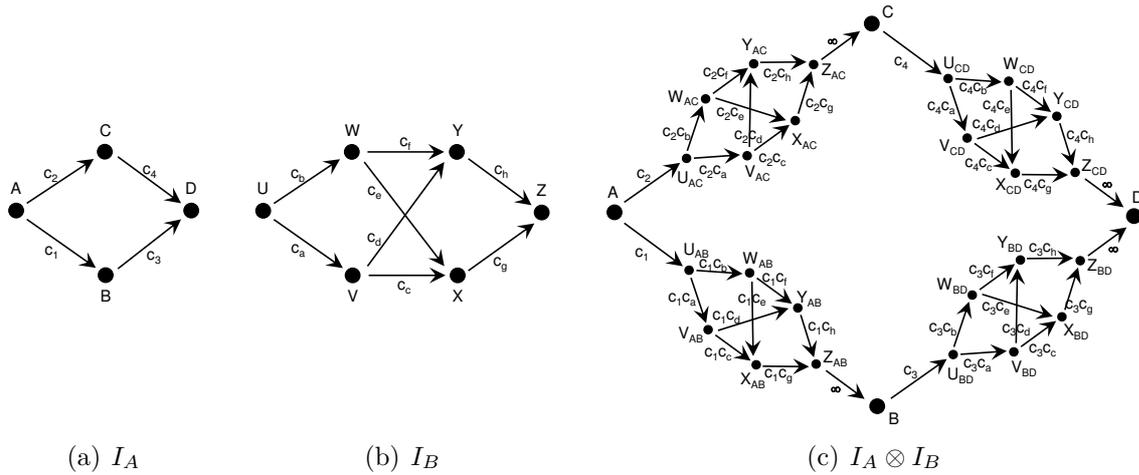
Először bevezettem a \otimes (kompozíció) operátort az MVPGS példányokra. Két példányból, I_A -ból és I_B -ből, egy új $I = I_A \otimes I_B$ példány készíthető kompozícióval, ha a következő feltételek teljesülnek:

1. I_B -ben az átvinni kívánt igény forgalommenyisége 1, és
2. a virtuális linkek számának korlátja (R) megegyezik I_A -ban és I_B -ben.

¹⁰Minimal Error Virtual Resource Allocation–Peer-Global Optimization with Given Weights

¹¹Minimal Error VRA-PGO with Given Weights for a Single Demand

¹²Polynomial Time Approximation Scheme class



8. ábra. MVPGS kompozíció

A 8. ábrán egy példa látható a kompozícióra. A kapacitások a linkek mellett vannak feltüntetve és minden link súlya egy egységnyi. I_A -ban az igény: $A \rightarrow D : 1$, I_B -ben pedig: $U \rightarrow Z : 1$. Mindhárom példány esetében $R = 1$.

Bevezettem továbbá az $OPT(I)$ jelölést, amely egy I MVPGS példány optimális megoldásának a mérőszámát jelenti, azaz a minimális β -t. Bevezettem a következő jelölést is:

$$\begin{aligned}
 I_0 &= \otimes^0 I = I \\
 I_1 &= \otimes^1 I = I \otimes I \\
 &\vdots \\
 I_k &= \otimes^k I = I \otimes (\otimes^{k-1} I)
 \end{aligned}$$

Ezután teljes indukcióval bebizonyítottam a következőt [Disszertáció, 30. lemma]:

Lemma. *Legyen I egy MVPGS példány, amelyre $OPT(I) \geq 1$. Ekkor bármely $k \in \mathbb{Z}$, $k \geq 0$ -ra: $OPT(\otimes^k I) = (OPT(I))^{k+1}$.*

Disszertációmiban ezt a lemmát, valamint a 26. és 28. tételt felhasználva bebizonyítottam a 29. tételt, ami megegyezik e tézis állításával. Más szóval ez azt jelenti, hogy az MVPGS feladat nem része az APX osztálynak.

4.3.4. Numerikus vizsgálat

Az előző, bonyolultságelméleti eredményeim szerint a globális egyenrangú optimalizálás általában véve nehéz feladat. Ez ugyanakkor nem jelenti azt, hogy a gyakorlatban előforduló hálózatokra nem létezik hatékony megoldás. Készítettem ezért egy számítógépes szimulációs rendszert, hogy megvizsgálhassam, hogy valóságghű környezetekben hogyan teljesítenek a tárgyalt algoritmusok. A szimulátor egy linkkapacitásokkal ellátott hálózatot és egy igényhalmazt vár bemenetként, majd több különböző algoritmussal megoldja a virtuális

erőforrás-foglalás feladatát. A szimulációs rendszert és az optimalizáló algoritmusokat C++ nyelven implementáltam, a LEMON Graph Library [16] használatával. A beágyazott lineáris programokat az IBM ILOG CPLEX Optimizer [17] segítségével oldottam meg.

A következő hét optimumkereső módszert valósítottam meg a szimulátoromban:

1. Átfedő optimalizálás, lásd a 4.1. alfejezetben.
2. Átfedő optimalizálás útvonalkizárással. Ez az átfedő optimalizálásnak egy kissé módosított változata, amelyben megengedjük az egyes útvonalak letiltását (vagyis $e_i = 0$ lehetséges).
3. Helyi egyenrangú optimalizálás a 4.2.3 alfejezetben bemutatott ILP-alapú iteratív megoldással.
4. Heurisztikus helyi egyenrangú optimalizálás, lásd a 4.2.3 alfejezet végén.
5. Globális egyenrangú optimalizálás, lásd a 4.3.2. alfejezetben.
6. Globális optimalizálás, lásd alább.
7. OSPF linksúly-optimalizálás (avagy OSPF-TE), lásd alább.

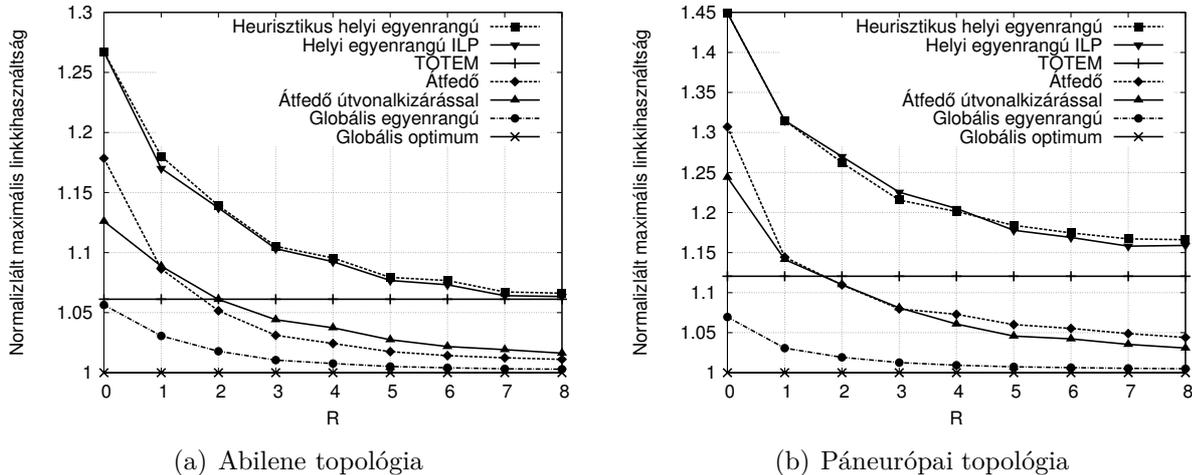
Ez utóbbi esetben a feladat az OSPF linksúlyokat úgy beállítani, hogy azokat átadva egy hagyományos OSPF-ECMP útvonalválasztásnak, a kialakuló hálózati forgalomban a maximális linkkihasználtság a lehető legkisebb legyen. Ebben az esetben egyáltalán nem alkalmazunk virtuális erőforrásokat. E feladról bizonyított, hogy NP-nehéz [5], ám ugyan ebben a cikkben a szerzők javasolnak egy heurisztikát is, amelyet a TOTEM (TOolbox for Traffic Engineering Methods [14, 15, 21]) nevű nyílt forráskódú rendszerben implementáltak. Ezt beemeltem a szimulátoromba, mint az OPSF linksúly-optimalizálásnak a jelenlegi legjobb gyakorlatát.

Röviden a *globális optimalizálásról*. Tekintsünk egy hálózatot linkkapacitásokkal és igényekkel. Ez egy klasszikus többtermékes folyamprobléma, melynek létezik lineáris programon alapuló megoldása, amely megadja az optimális linkenkénti és igényenkénti forgalommenyiségeket. Ha egy kellően kifinomult forgalomelvezetési mechanizmus segítségével az igényeket pontosan e megoldásnak megfelelően tudnánk elvinni, akkor az elméletileg lehetséges legkisebb maximális linkkihasználtsághoz juthatnánk. Éppen ezért bevettem egy egyszerű algoritmust a szimulátoromba, amely megoldja ezt a többtermékes folyamproblémát leíró LP-t, és a kimenetet valódi forgalommenyiségeknek tekinti. Ezek az eredmények azután referenciaként fognak szolgálni, hiszen semmilyen algoritmus (sem átfedést használó, sem egyenrangú) nem teljesíthet ennél jobban. Sőt tovább mentem, ugyanis a különböző algoritmusok által eredményezett maximális linkkihasználtságokat elosztottam ezzel az optimális maximális linkkihasználtsággal, hogy ily módon normalizált értékekhez jussak, amelyek függetlenek a linkek sávszélességeitől és a forgalommenyiségektől. Ennek az optimalizálásnak az eredménye *Globális optimum*-ként van feltüntetve a grafikonokon. Természetesen az említett normalizálás miatt a Globális optimum maximális linkkihasználtság értékei konstans 1,0-k.

Disszertációmban részletesen bemutattam a szimulációs eredményeimet, itt csak a legérdekesebbeket emeltem ki. Ezeket a válogatott eredményeket két valóság-hű hálózati topológia szimulálásával nyertem: az egyik a jól ismert észak-amerikai Abilene hálózat, a másik pedig egy egyszerűsített páneurópai optikai gerinchálózat (ld. a 4. táblázatot). Mindkét esetben minden link sávszélessége 100 egységre volt állítva. Minden szimulációs menetben 5 igény szerepelt és minden esetben a forrás- és célsomópontokat véletlenszerűen választottam. Az

Hálózat	Csomópontok száma	Egyirányú linkek száma	Linkkapacitások [egység]
Abilene	12	30	100
Páneurópai	16	46	100

4. táblázat. A hálózatok jellemzői



9. ábra. A szimulációs eredmények

igényelt forgalommenyiségeket is véletlenszerűen sorsoltam, minden igényhez egyenletes eloszlással az $[5, 30]$ egység intervallumból. A virtuális linkek vagy útvonalak megengedett maximális számát (R) 0 és 8 között változtattam, a határokat is beleértve. A szimulációkat 300-szor ismételt meg – mind a hét algoritmussal – egy nagy teljesítményű számítógépen, hogy csökkenjen az eredmények szórása.

A legérdekesebb eredmények a 9. ábrán láthatóak. A TOTEM majdnem olyan jól szerepelt, mint a globális egyenrangú optimalizálás a virtuális linkek nélküli esetben, ami a TOTEM elvi alsó korlátja. Az $R = 0$ esetben a helyi egyenrangú és az átfedő optimalizálások egyértelműen rosszabbul teljesítettek a TOTEM-nél, ami nem meglepő, hiszen egy virtuális erőforrás-foglaló algoritmust virtuális linkek vagy útvonalak nélkül futtatni nem túlzottan értelmes. Másrészt viszont ha csak két vagy három virtuális linket megengedünk csomópontonként, akkor az átfedő optimalizálás már egyértelműen túlszárnyalja a TOTEM-et, és pár százalékra megközelíti a globális optimumot is. A helyi egyenrangú ILP és helyi egyenrangú heurisztika teljesítménye volt a leggyengébb, de ezek csak gyors heurisztikák a VRA-PGO feladatra. A globális egyenrangú optimalizálás maximális linkkihasználtsága jóval a TOTEM-é alatt van már $R = 1$ esetében, és tovább csökken ahogy R nő, majdnem elérve a globális optimumot már $R = 4$ -nél. Mindez azt mutatja, hogy a globális egyenrangú optimalizálásban nagy lehetőségek vannak, de a jelenleg alkalmazott heurisztikák nem képesek ezt kellően kiaknázni, így jobb algoritmusok keresése jövőbeli kutatások tárgya lehet.

Ami különböző algoritmusok *erőforrásigényét* illeti, nagyon rövid (kb. másodperces vagy az alatti) futásidőket mértem az átfedő, a helyi egyenrangú és a globális optimalizálásoknál. Ez azt jelenti, hogy ezek az algoritmusok valószínűleg használhatóak olyan környezetben,

ahol rövid válaszidők szükségesek, mint például egy forgalomelvezetés valós idejű optimalizálása. A TOTEM esetében a számítások több tíz másodpercig tartottak, akár a perc hosszúságot is meghaladva, ami azonban még mindig jó lehet nem valós idejű feladatokra. Ezekben az esetekben a memóriefoglalás is szerény volt, alig 4–10 MB. Ugyanakkor a globális egyenrangú optimalizálásnál az átlagos futásidők több óra nagyságrendjébe emelkedtek. Ez esetben a szórás is sokkal nagyobb volt: egy szimulációs menet futási ideje a néhány másodperctől a több napig tartó skálán mozgott. A memóriaigény szintén erősen változó volt, néhány MB-tól majdnem 30 GB-ig terjedt. Mindez azt jelenti, hogy a javasolt globális egyenrangú optimalizálási módszer sok gyakorlati esetben valószínűleg nem használható.

5. Az új eredmények alkalmazhatósága

Disszertációmban bevezettem és részletesen megvizsgáltam a virtuális erőforrás-foglalás koncepcióját, amelynek bemutatására munkám nagy részében az OSPF fogalomelvezetés feladatát használtam példaként. Így természetesen a bemutatott optimalizációs módszerek és algoritmusok közvetlenül felhasználhatóak OSPF forgalomelvezetést alkalmazó valós kommunikációs hálózatokban, növelve e hálózatok teljesítőképességét.

Egy másik lehetséges alkalmazási terület a WCMP protokoll tökéletesítése. A WCMP-t bemutató [2] konferenciacikk 3.2.1–3.2.3 alfejezeteiben a szerzők csupán heurisztikus megoldásokat javasolnak a szabályoptimalizálásra. Az 1.3 tézisben javasolt algoritmusaim közvetlenül alkalmazhatók a WCMP feladataira is, ráadásul azok mindig optimális megoldást biztosítanak, nem csak közelítést. Ehhez kapcsolódik disszertációm 3.4.1 alfejezete, amely a VRA WCMP-ben való alkalmazásáról szól.

További felhasználási terület lehet például a Fibbing rendszer [8]. Ebben, ha egy útvonalválasztásban egyetlen legrövidebb út van egy célsomóponthoz, de mégis két párhuzamos útvonalat kívánunk használni, közöttük egyenletesen elosztott forgalommal, akkor egy nem létező csomópontot és egy nem létező linket kell meghirdetni a hálózatban. Ugyanígy, ha például 33%–67% arányú forgalomosztást szeretnénk elérni, akkor két nem létező csomópontot és linket kell hirdetni. Ebből is látszik, hogy a VRA rendszerem jól használható Fibbing környezetben tetszőleges osztásarány legjobb közelítésének megtalálására korlátos számú nem létező csomópontról és linkről szóló bejegyzéssel.

A COYOTE [22] egy frissen javasolt forgalomelvezetési rendszer, amely megenged némi bizonytalanságot a forgalmi igények megadásakor. Ebben a szerzők a Fibbing ötletét alkalmazzák kombinálva néhány VRA-1N-1D-hez készített algoritmusokkal, hogy így közelítsék meg a forgalomelvezetés optimumát az általuk vizsgált esetben.

Ahogy ezek a példák is mutatják, a virtuális erőforrás-foglalás sok olyan helyen használható egy hálózatban, ahol optimális erőforrás-megosztást szeretnénk elérni hagyományos egyenletesen osztó rendszerek használatával.

Végül megemlítem, hogy készítettünk Kőrösi Attila társszerzőmmel egy szabadalmi előterjesztést a virtuális erőforrás-foglalásról. Az Ericsson nemzetközi távközlési készülékgyártó cég értékesnek találta ezt és megvásárolta, majd benyújtotta a szabadalmi hivatalokhoz, úgy, hogy a feltalálók mi maradtunk, de az Ericsson a bejelentő [O1]. Mindez szintén az itt bemutatott munkámnak a fontosságát és hasznosíthatóságát támasztja alá.

Hivatkozások

- [1] D. Wischik, M. Handley, and M. B. Braun, „The resource pooling principle,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 5, pp. 47–52, 2008.
- [2] J. Zhou, M. Tewari, M. Zhu, A. Kabbani, L. Poutievski, A. Singh, and A. Vahdat, „WCMP: Weighted cost multipathing for improved fairness in data centers,” in *Proceedings of the Ninth European Conference on Computer Systems*, ser. EuroSys, Apr. 2014, pp. 5:1–5:14.
- [3] J. T. Moy, „OSPF Version 2,” RFC 2328, Mar. 2013.
- [4] International Organization for Standardization, „Intermediate System to Intermediate System intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473),” ISO/IEC 10589:2002, November 2002.
- [5] B. Fortz and M. Thorup, „Increasing internet capacity using local search,” *Computational Optimization and Applications*, vol. 29, pp. 13–48, 2004.
- [6] N. Kang, M. Ghobadi, J. Reumann, A. Shraer, and J. Rexford, „Efficient traffic splitting on commodity switches,” in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT ’15, Dec. 2015, pp. 6:1–6:13.
- [7] A. Maghbouleh, „Metric-based traffic engineering: Panacea or snake oil? A real-world study,” in *27th North American Network Operators Group Meeting (NANOG27)*, Feb 2003.
- [8] S. Vissicchio, O. Tilmans, L. Vanbever, and J. Rexford, „Central Control Over Distributed Routing,” in *ACM SIGCOMM*, August 2015, pp. 43–56.
- [9] O. Tilmans, S. Vissicchio, L. Vanbever, and J. Rexford, „Fibbing in action: On-demand load-balancing for better video delivery,” in *ACM SIGCOMM*, 2016, pp. 619–620.
- [10] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, „Overview and principles of Internet traffic engineering,” RFC 3272, May 2002.
- [11] Y. Wang and Z. Wang, „Explicit routing algorithms for internet traffic engineering,” in *Computer Communications and Networks, 1999. Proceedings. Eight International Conference on*, 1999, pp. 582–588.
- [12] Z. Wang, Y. Wang, and L. Zhang, „Internet traffic engineering without full-mesh overlaying,” in *IEEE INFOCOM 2001*, vol. 1, April 2001, pp. 565–571.
- [13] M. Chiesa, G. Kindler, and M. Schapira, „Traffic Engineering with Equal-Cost-MultiPath: An Algorithmic Perspective,” in *IEEE INFOCOM 2014*, April 2014, pp. 1590–1598.

- [14] G. Leduc, H. Abrahamsson, S. Balon, S. Bessler, M. D'Arienzo, O. Delcourt, J. Domingo-Pascual, S. Cerav-Erbas, I. Gojmerac, X. Masip, A. Pescapè, B. Quoitin, S. P. Romano, E. Salvadori, F. Skivée, H. T. Tran, S. Uhlig, and H. Ümit, „An open source traffic engineering toolbox,” *Comput. Commun.*, vol. 29, no. 5, pp. 593–610, Mar. 2006.
- [15] H. Ümit, „Interior Gateway Protocol Weight Optimization Tool,” Feb. 2007, letöltve 2018 nov. [Online]. Cím: <http://www.poms.ucl.ac.be/totem/>
- [16] „LEMON Graph Library – Library for Efficient Modeling and Optimization in Networks,” July 2014, Version 1.3.1. Letöltve 2018 nov. [Online]. Cím: <http://lemon.cs.elte.hu/>
- [17] „IBM ILOG CPLEX Optimizer,” Letöltve 2018 nov. [Online]. Cím: <https://www.ibm.com/analytics/cplex-optimizer>
- [18] „Maple (mathematical software),” 2018, letöltve 2018 nov. [Online]. Cím: <http://www.maplesoft.com/products/Maple/>
- [19] R. Joseph, *Galois Theory*, ser. Universitext. Springer New York, 1998, second edition.
- [20] V. Kann, „On the Approximability of NP-complete Optimization Problems,” Ph.D. dissertation, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, May 1992.
- [21] „TOTEM Project: TOolbox for Traffic Engineering Methods,” Latest version (3.2.1) released at Nov. 2008. Letöltve 2018 nov. [Online]. Cím: <http://totem.run.montefiore.ulg.ac.be/>
- [22] M. Chiesa, G. Rétvári, and M. Schapira, „Lying your way to better traffic engineering,” in *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '16, 2016, pp. 391–398.

Publikációk

Nemzetközi lektorált folyóiratcikkek

- [J1] **Krisztián Németh**, Attila Kőrösi, Gábor Rétvári. „Optimal Resource Pooling over Legacy Equal-Split Load Balancing Schemes”, *Computer Networks*, 127, Nov. 2017, pp. 243-265. (6/2=3 pont)
- [J2] Nikolett Bereczky, Amalia Duch, **Krisztián Németh**, Salvador Roura. „Quad- kd trees: A general framework for kd trees and quad trees”, *Theoretical Computer Science*, 616, Feb. 2016, pp. 126-140. (6/4=1,5 pont)
- [J3] Péter Füzesi, **Krisztián Németh**, Niklas Borg, Rikard Holmberg, István Cselényi. „Provisioning of QoS enabled inter-domain services”, *Computer Communications*, 26 (10), Jun. 2003, pp. 1070-1082. (6/5=1,2 pont)
- [J4] Gábor Fehér, **Krisztián Németh**, István Cselényi. „Performance Evaluation Framework for IP Resource Reservation Signaling”, *Performance Evaluation*, 48 (1-4), May 2002, pp. 131-156. (6/3=2 pont)

Magyarországon megjelent folyóiratcikkek

- [J5] **Németh Krisztián**. „Hívásengedélyezés garantált minőségű hálózatokban (áttekintés)”, *Híradástechnika*, ISSN: 0018-2028, LVII, 2002/9, pp. 2-4. (Lektorált, 2 pont)
- [J6] Cselényi István, Füzesi Péter, **Németh Krisztián**. „Az internet szolgáltatásminőség fejlődése”, *Magyar Távközlés*, ISSN: 0865-9648, 2000/2, pp. 26-31. (Nem lektorált, 1/3=0,333 pont)
- [J7] **Krisztián Németh**. „IP Multicasting over ATM”, *Magyar Távközlés, Selected Papers*, ISSN 0865-9648, 1999, pp. 11-15. (Nem lektorált, idegen nyelvű, 2 pont)
- [J8] **Németh Krisztián**. „IP multicast ATM felett”, *Magyar Távközlés*, ISSN: 0865-9648, 1998/7, pp. 3-6. (Nem lektorált. A [J7] ennek az átdolgozott fordítása, 0 pont)

Nemzetközi konferenciapublikációk

- [C1] **Krisztián Németh**, Attila Kőrösi, Gábor Rétvári. „Enriching the poor man’s traffic engineering: Virtual link provisioning for optimal OSPF TE”, *Networks 2014*, Funchal, Madeira Island, Portugal, September 2014. (3/2=1,5 pont)
- [C2] Nikolett Bereczky, Amalia Duch, **Krisztián Németh**, Salvador Roura. „Quad- K - d Trees” *Latin American Theoretical INformatics (LATIN 2014)*, Montevideo, Uruguay, March 31 - April 4, 2014, Proc. LNCS 8392, pp. 743-754. (3/4=0,75 pont)

- [C3] **Krisztián Németh**, Attila Kőrösi, Gábor Rétvári. „Optimal OSPF Traffic Engineering using Legacy Equal Cost Multipath Load Balancing”, *IFIP Networking 2013*, Brooklyn, New York, USA, May 2013 (3/2=1,5 pont)
- [C4] Tibor Cinkler, Réka Kosznai, Péter Balázs Soproni, **Krisztián Németh**. „GSP, the Generalised Shared Protection” *9th International Conference on Design of Reliable Communication Networks (DRCN 2013)*, Budapest, Hungary, March 2013. (3/4=0,75 pont)
- [C5] **Krisztián Németh**, Gábor Rétvári. „Traffic Splitting Algorithms in Multipath Networks: Is the Present Practice Good Enough?”, *Networks 2012*, Rome, Italy, October 2012. (3 pont)
- [C6] Zsolt Kovácsházi, Gábor Papp, **Krisztián Németh**. „A Hybrid Multicast Video Distribution Method: a Technology for E-Entertainment”, *2005 Networking and Electronic Commerce Research Conference (NAEC 2005)*, Riva del Garda, Italy, October 2005, pp. 1-11. (3/3=1 pont)
- [C7] Niklas Borg, Rikard Holmberg, Péter Füzesi, **Krisztián Németh**. „NAIS – Network Architecture for Inter-Domain Services”, *Networks 2002*, Munich, Germany, June 2002. Proceedings ISBN: 3-8007-2711-0 (3/4=0,75 pont)
- [C8] **Krisztián Németh**, Péter Füzesi. „An Analysis of IP Resource Reservation Protocols”, *Networks 2002*, Munich, Germany, June 2002. Proceedings ISBN: 3-8007-2711-0, pp. 213-220 (3/2=1,5 pont)
- [C9] István Moldován, **Krisztián Németh**. „Quality of Service Architectures Using MPLS Networks”, *9th IFIP Working Conference on Performance Modelling and Evaluation of ATM and IP Networks*, Budapest, Hungary, June 2001. Proceedings ISBN 963 420 694 8 (3/2=1,5 pont)
- [C10] István Moldován, **Krisztián Németh**, Tibor Cinkler. „Merging in MPLS Networks”, *IEEE ICT 2001*, Bucharest, Romania, June 2001 (3/3=1 pont)
- [C11] Csaba Simon, **Krisztián Németh**, Sándor Székely. „Point-to-Multipoint ATM Signalling Performance Measurements”, *8th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, Ilkley, UK, July 2000. Proceedings: Networks UK, D. D. Kouvatsos (Ed.) ISBN 0-9540151-1-8 (3/3=1 pont)
- [C12] **Krisztián Németh**, Gábor Fehér, István Cselényi. „Simulation Study for IP Resource Reservation”, *8th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, Ilkley, UK, July 2000. Proceedings: Networks UK, D. D. Kouvatsos (Ed.) ISBN 0-9540151-1-8 (3/3=1 pont)
- [C13] Gábor Fehér, **Krisztián Németh**, István Cselényi. „Router Benchmarking Framework for QoS Signaling”, *8th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, Ilkley, UK, July 2000. Proceedings: Networks UK, D. D. Kouvatsos (Ed.) ISBN 0-9540151-1-8 (3/3=1 pont)

- [C14] István Cselényi, Gábor Fehér, **Krisztián Németh**. „Benchmarking of Signaling Based Resource Reservation in the Internet”, *Networking 2000*, IFIP-TC6, Paris, France, May 2000. Proceedings: LNCS 1815; Networking 2000: Broadband Communications, High Performance Networking and Performance of Communication Networks; ed. Guy Pujolle et al., ISSN 0302-9743, ISBN 3-540-67506-X (3/3=1 pont)
- [C15] **Krisztián Németh**, Krzysztof Szarkowicz. „IP Multicasting over ATM”, *7th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, IFIP WG 6.3, 6.4, Antwerp, Belgium, June 28-30, 1999 (3/2=1,5 pont)
- [C16] Gábor Fehér, **Krisztián Németh**, Markosz Maliosz, István Cselényi, Joakim Bergkvist, David Ahlard, Tomas Engborg. „Boomerang – A Simple Protocol for Resource Reservation in IP Networks”, *IEEE Workshop on QoS Support for Real Time Internet Applications in conjunction with Real-Time Technology and Applications Symposium (RTAS)*, Vancouver, Canada, June 2-4, 1999 (3/7=0,429 pont)

Magyar nyelvű kiadványban megjelent konferenciaközlemény

- [C17] **Németh Krisztián**. „Hívásengedélyezés garantált minőségű csomagkapcsolt hálózatokban”, *PKI Tudományos Napok 111*, Budapest, November 2002. (1 pont)

Egyéb közlemények

- [O1] Attila Kőrösi, **Krisztián Németh**. „Methods and packet network devices for forwarding packet data traffic”, *International Patent Application*, Int. Application No.: PCT/EP2013/060404, International Filing Date: 21 May, 2013, Publication No.: WO/2014/187475, Publication Date: 27 Nov., 2014 (Külföldi szabadalom, 2/2=1 pont)
- [O2] Gábor Fehér, **Krisztián Németh**, András Korn, István Cselényi. „Benchmarking Terminology for Resource Reservation Capable Routers”, *IETF RFC 4883*, July 2007 (0 pont)

Publikációs összpontszám: 33,212