



Budapesti Műszaki és Gazdaságtudományi Egyetem
Távközlési és Médiainformatikai Tanszék

Eljárások Következő Generációs Mobil és Adat Hálózatok Tervezésére

Harmatos János

*Nagysebességű Hálózatok Laboratórium
Távközlési és Médiainformatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem*

Ph. D. Tézisek

Tudományos vezető:

Dr. Sallai Gyula

*Távközlési és Médiainformatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem*

Budapest
2003

1. Bevezetés

Napjainkban az informatika, illetve a távközlés világában jelentős változások mennek végbe, melyek hatással vannak a hálózatok tervezésére is. A klasszikus áramkörkapcsolás egyre inkább háttérbe szorul, továbbá az elkövetkezendő néhány évben a beszédátvitel részesedése várhatóan 50%-al fog csökkenni. Ugyanakkor az adatátvitel jelentősége és mennyisége folyamatosan nő, és az erre alapuló számos új szolgáltatás (Multimédia Üzenet Szolgáltatás (Multimedia Message Service - MMS), Mobil Internet hozzáférés és további értéknövelt szolgáltatások) elterjedése igényt támaszt egy *integrált, több-szolgáltatású* hálózati technológiára.

A mobil hálózatok területén végbemenő változások nagyon jól illusztrálják a fenti folyamatot. A GSM előfizetők számának folyamatos növekedése mellett a harmadik generációs (3G) mobil rendszerek (például az UMTS - Universal Mobile Telecommunication System) is megjelentek. Mivel a 3G rendszerek számos különböző szolgáltatás támogatására képesek, várhatóan fontos szerepet fognak játszani a távközlési piacon. Továbbá a 3G készülékek képesek lesznek arra, hogy egyetlen olyan eszközt adjanak az előfizetők kezébe, amellyel minden olyan szolgáltatáshoz hozzá tudnak férni, amelyek igénybevételéhez eddig több különböző készülékre volt szükség. A folyamatosan növekvő adatforgalom és az Internet Protokoll (IP) elterjedése miatt várhatóan a 3G törzs (gerinc) hálózatok úgynevezett *teljes-IP (all-IP)* alapú hálózatok lesznek. A hálózattervezés szempontjából nézve a harmadik generációs mobil hálózatok számos új, eddig még nem nagyon vizsgált tervezési feladatot vetnek fel mind a földi hozzáférési, mind pedig a törzs hálózattal kapcsolatban.

Az IP hálózatok széleskörű elterjedéséhez kapcsolódik az OSPF (Open Shortest Path First) útvonal-választó protokoll előtérbe kerülése. Az OSPF egy "összeköttetés-állapot" alapú útvonalválasztó protokoll, amely az összeköttetésekhez rendelt ún. adminisztratív súlyok alapján határozza meg legrövidebb utakat az útvonal-választási folyamatban. A hálózat minél jobb teljesítőképessége érdekében fontos ezen súlyok megfelelő beállítása.

A szükséges szolgáltatás-minőség elérése, illetve a forgalom-menedzsment alkalmazhatósága érdekében a 3G törzs hálózatokban várhatóan széles körben fogják alkalmazni a Több-Protokollos Címke Kapcsolást (Multi-Protocol Label Switching - MPLS). Az MPLS elterjedése hatással van a nagysebességű, országos méretű gerinchálózatokra is, ahol jelenleg a Hullámhossz Osztásos Nyalábolás (Wavelength Division Multiplexing - WDM) a legelterjedtebb módja a forgalom továbbításának. Az Európai Távközlési Szabványosítási Hivatal (European Telecommunication Standardization Institute - ETSI), illetve az Optikai Hálózat-együttműködési Fórum (Optical Internetworking Forum) komoly erőfeszítéseket tesz annak érdekében, hogy az MPLS alapvetően optikai hálózatokban is alkalmazni lehessen. Ezen törekvés eredményeképpen született meg az ún. Általánosított Több-Protokollos Címke Kapcsolás (Generalized Multi-Protocol Label Switching - GMPLS), illetve az ún. Több-Protokollos Hullámhossz Kapcsolás (Multi-Protocol Lambda Switching - MPLS). E tématerület fontos nyitott kérdése olyan

erőforrás-foglalási stratégia kidolgozása, amely jól illeszkedik az adatforgalom dinamikus, borsztös jellegéhez.

A fentebb említett tendenciák, az új hálózati technológiák fókuszba kerülése számos megoldásra váró hálózattervezési illetve optimalizációs problémát vetnek fel.

A 3G hálózatok esetében meg kell oldanunk mind a hozzáférési, mind a törzs hálózat topológiájának megtervezését, amely magában foglalja a kapcsoló eszközök helyének meghatározását, illetve ezen eszközök összekötését. Mivel a topológia nagyban befolyásolja az építési és a hosszútávú fenntartási költségeket, a megfelelő topológiai tervezés az egyik legkritikusabb pont a szolgáltató versenyképessége szempontjából. A topológia tervezés mellett az igények optimális úton való elvezetése is fontos tervezési feladat. Mivel a 3G hálózatok topológiája technikai okokból kifolyólag számos kötöttséggel rendelkezik, továbbá a hálózat forgalmának speciális tulajdonságai vannak, ezért a rendelkezésünkre álló tervező algoritmusok nem alkalmazhatók; ha megfelelő eredményt szeretnénk elérni új tervezési módszerekre van szükség.

Az IP alapú hálózatok folyamatos terjedése felszínre hozta azt a problémát, hogy a jelenleg általánosan alkalmazott OSPF adminisztratív súly beállító megoldások nem eredményeznek kellően jó hálózat kihasználtságot. A szolgáltatók részéről felmerült az igény olyan hatékony módszerekre, amelyek képesek olyan OSPF súlyrendszert eredményezni, amely biztosítja az IP hálózat megfelelő kihasználtságát. Mivel ez egy új kutatási terület, ezért még kevés számottevő eredmény született e témában.

A WDM, illetve az MPLS hálózatok esetében az úgynevezett dinamikus hullámhossz és útvonal foglalási technikák kerültek a kutatások középpontjába. Jónéhány egyszerű, robusztus stratégiát javasoltak már eddig, de az ezek által szolgáltatott eredmények messze vannak az optimális megoldástól, bonyolultabb módszerekkel várhatóan számottevően jobb eredmény érhető el.

Minden fenti esetben a feladatok komplexitása, a hálózatok nagy mérete és bonyolult felépítése, valamint a forgalom jelentős növekedése ún. *heurisztikus algoritmus alapú* optimalizálást tesz szükségessé a hatékony hálózattervezéshez.

2. Kutatási célkitűzések

A Bevezetésben már utaltam rá, hogy napjaink és a közeljövő hálózataiban számos megoldásra váró tervezési probléma merül fel. Doktori értekezésemben olyan hatékony hálózattervezési *eljárásokat* és *algoritmusokat* javasolok, amelyek eredményesen használhatók a következő generációs mobil, illetve IP/OSPF és optikai hálózatokban. Munkám során az alábbi feladatokkal foglalkoztam:

- UMTS hozzáférési hálózat költség alapú tervezése: Céлом egy olyan összetett tervezési eljárás kifejlesztése volt, amely képes a fenti feladatot elfogadható futási időn belül, hatékonyabban megoldani, mint a rendelkezésre álló módszerek.

- UMTS törzs hálózat költség alapú tervezése: Javasoltam egy globális heurisztikus algoritmust és egy lokális javító eljárást, amelyek együttműködve költség takarékosabb hálózatot képesek tervezni, mint a jelenleg ismert módszerek.
- OSPF adminisztratív súlyok optimalizálása: Olyan algoritmikus súlybeállító módszerek kidolgozása volt a céлом, amelyek képesek jobb hálózat kihasználtságot eredményező súlyrendszert találni, mint a jelenleg alkalmazott megoldások.
- Dinamikus útvonal és hullámhossz választás WDM hálózatokban: Olyan stratégiákat javasoltam dinamikus útvonal és hullámhossz választásra optikai hálózatokban, amelyek kisebb blokkolást eredményeznek, mint az eddig használt algoritmusok.

Az algoritmusok kidolgozásánál az alábbi célokat tűztem ki, valamint az alábbi feltételeket és követelményeket vettem figyelembe:

- Olyan tervezési módszereket javasoltam, amelyek *gyorsabbak* vagy/és *jobb* eredményeket érnek el, mint a létező algoritmusok
- Olyan feladatoknál, ahol a futási idő nem kritikus (pl. zöld mezős hálózat építés) az algoritmusok kidolgozása során a minél jobb eredmény elérése volt a cél, a futási idő hossza nem volt elsődleges szempont.
- Valós idejű (on-line) feladatoknál az eredmény minősége mellett a gyors működés az alapvető tényező.

Bár mindegyik algoritmus adott hálózatban felmerülő feladat megoldására készült, de képesek kezelni a különböző típusú hálózatokban megjelenő hasonló tervezési feladatokat jelentősebb módosítások nélkül (technológia függetlenség).

3. Kutatási módszerek

A távközlési hálózatokat gráfokkal modelleztem. A felvázolt tervezési problémák megoldásában felhasználtam az alábbi, jól ismert gráfelméleti algoritmusokat:

- Dijkstra és Suurballe algoritmusok a minimális költségű legrövidebb utak illetve útpárok megkeresésére.
- Prim és Kruskal algoritmus a minimális költségű feszítőfák meghatározására.

A fenti algoritmusokat, mint alapelemeket használva olyan tervezési eljárásokat dolgoztam ki, amelyek képesek megoldani a 2. fejezetben vázolt tervezési problémákat.

A legtöbb felmerülő probléma olyan *optimalizálási feladat*, amelyeket megfogalmazhatunk *lineáris programozási* vagy *egészértékű lineáris programozási* feladatként. Az ilyen feladatok megoldhatók a rendelkezésre álló programcsomagokkal, (ilyenek például a CPLEX, illetve az LP-solve). Ily módon meghatározható a feladat *optimális* megoldása, de a gyakorlatban előforduló, nagyobb méretű hálózatokra a problémater nagysága miatt ez a módszer elfogadható időn belül nem ad megoldást.

Emiatt heurisztikus algoritmusok kifejlesztésére törekedtem, amelyek megállják helyüket a gyakorlatban felmerülő tervezési problémák esetében. Az általam kidolgozott

tervezési eljárások egy részében ún. *általános heurisztikus algoritmusokat* alkalmaztam, mint amilyen a *Szimulált Lehűtés - Simulated Annealing (1. és 2. tézisek)*, a *Szimulált Helyfoglalás - Simulated Allocation (2. tézis)*, és a *Genetikus Algoritmus - Genetic Algorithm (1. tézis)*. A lehető legjobb eredmény elérése érdekében az előbbi algoritmusokat egyrészt illesztettem az adott tervezési feladathoz, másrészt több ponton továbbfejlesztettem. Számos tervezési eljárásban alkalmaztam általam kidolgozott *probléma specifikus heurisztikákat (2.,3.,4. tézisek)*, amelyeknél teljes mértékben figyelembe vettem az adott optimalizálási feladat jellegét, így ezek a módszerek képesek arra, hogy jobb eredményeket érjenek el, mint az általános eljárások.

Az algoritmusok teljesítményének kiértékeléséhez a *szimulációt* használtam, ami általánosan használt módszer. Választásom oka a tárgyalt problémák komplexitása és a nagy állapottér volt, amelyek miatt az analitikus kiértékelés nem, vagy csak nagyon korlátozottan alkalmazható. Azokban az esetekben ahol lehetséges volt alsó becslést adni az optimális megoldásra, ezt a fajta kiértékelési módot is használtam.

4. Új eredmények

1. tézis: Költség alapú UMTS földi rádiós hozzáférési hálózatok (UMTS Terrestrial Radio Access Network – UTRAN) tervezése [C1, C4, C8, J6]

Az UMTS földi hozzáférési hálózatának tervezése magában foglalja a Rádiós Hálózatvezérlők (Radio Network Controller - RNC) elhelyezésének, valamint a Rádiós Bázisállomások (Radio Base Station - RBS) egymással és az RNC-vel való összekötésének problémáját. A cél egy költség optimális többkorlátos fa szerkezetű topológia kialakítása. A feladat egyik nehézsége a bonyolult költség struktúra, mivel a hálózat teljes költsége két részből (RNC-k számától függő költség, RBS-ek összekötéséből adódó költség) tevődik össze, amelyek egymásra is hatással vannak. További probléma, hogy a fa topológiának fokszám és mélység korlátbeli megkötései is vannak. A szerzők [5]-ben megmutatták, hogy a több korlátos fa tervezése önmagában \mathcal{NP} -teljes probléma. Ugyancsak nehézséget jelent, hogy az összeköttetések költsége mind a távolságtól, mind a kapacitástól függ, valamint a bázisállomások is rendelkeznek kapacitástól függő, lépcsős jellegű költségfüggvénnyel. [9]-ben a szerzők javasolnak egy algoritmust mobil hozzáférési hálózatok tervezésére, de a megtervezendő topológia itt egyszerű fa vagy csillag. [C1]-ben egy összetett algoritmus került ismertetésre, amely egy fázisban oldja meg a fent vázolt problémákat.

Javasolt módszeremben abból indultam ki, hogy ha az RNC-k helyének meghatározását, illetve a fa topológia tervezését két lépésben, különálló algoritmusok segítségével oldom meg, majd ezeket kombinálom, akkor jobb eredményt tudok elérni.

A tervezési probléma két részre bontásának megfelelően 1.1 és 1.2 altézisemben egy – egy algoritmust javasoltam, amelyek az alábbi feladatokat oldják meg:

- 1. feladat: Az RNC-k számának és helyének optimális meghatározása. A feladat

része annak a meghatározása is, hogy melyik RBS melyik RNC-hez kapcsolódjon a későbbiekben. (1.1 altézis)

- 2. feladat: A fa topológiájú RBS hálózat megtervezése. (1.2 altézis)

A tézis végén röviden összehasonlítom a javasolt algoritmusok és néhány ismert tervezési módszer teljesítőképességét.

1.1. altézis: 1. feladat megoldása – RNC helymeghatározó algoritmus (RNC Localization Algorithm - RLA)

Új, továbbfejlesztett genetikus algoritmusra épülő tervezési eljárást javasoltam, amely képes az RNC-k számának és helyének költség optimális meghatározására, illetve megoldja az RBS-ek RNC-khez történő hozzárendelésének problémáját.

A javasolt tervezési eljárás a genetikus algoritmus probléma specifikus alkalmazásán alapszik. A genetikus algoritmus a természetben is megfigyelhető biológiai változásokat modellezi. Hasonlóan a természetben megfigyelhető szelekcióhoz az egyedek között, a genetikus algoritmus is képes arra, hogy egyre jobb minőségű eredményeket érjen el iterációnként, feltéve ha a problémátér jól van kódolva. Az általános genetikus algoritmus az alábbi operátorokat használja: rekombináció, keresztezés és mutáció.

Az általam javasolt algoritmus főbb lépései a következők:

Kezdeti állapot generálása. Modellemben az egyed egy x vektor, amely annyi elemet tartalmaz, ahány lehetséges helyre lehet RNC-t elhelyezni. Ha a vektor i -edik eleme, $x_i = 1$, akkor van RNC az adott helyen, különben nincs. Miután az egyedek száma beállításra került, véletlenszerűen elhelyezünk RNC-ket a populáció minden egyedében.

1. lépés Rekombináció. Itt alkalmaztam az ún. "elitizmus" technikáját, amely azt jelenti, hogy mindig az előző populáció két legjobb egyede lesz kiválasztva (szülőknek), és az új populáció egyedei belőlük lesznek létrehozva a keresztezés operátor által.

2. lépés Keresztezés. A klasszikus keresztesési operátort továbbfejlesztettem, oly módon, hogy bevezettem egy *best* (legjobb) nevű extra egyedet, amely független a populációtól. Ezen egyed feladata, hogy az optimalizálás eddigi lépéseiről tároljon információt, amelyet minden keresztezésnél figyelembe veszünk. A *best* vektor egy pozíciója azt a számot tárolja, ahányszor egy RNC-t elhelyeztünk oda az eddigi iterációk alatt. Az általam javasolt keresztezés operátor esetében ha a két szülő vektor azonos pozíciójában különböző értékek vannak, akkor a *best* egyed megfelelő értékét is figyelembe vesszük az alábbiak szerint. Adott a valószínűsége annak, hogy az új egyed attól a szülőtől örököl, amelynek az értéke többször fordul elő a *best* vektorban (jelöljük ezt a valószínűséget P -vel). Jelöljük a jelenlegi lépés számát j -vel, a *best* egyed i -edik helyén lévő számot c -vel, továbbá adott egy 0 és 1 közötti véletlenszám (R -el jelöljük). Arról, hogy melyik szülőtől örököljön az új egyed a következőképpen dől el:

- Ha $R \leq P$ és $c > (j - c)$ akkor az új egyedben $x_i = 1$ lesz, egyébként 0.
- Ha $R > P$ és $c > (j - c)$ akkor az új egyedben $x_i = 0$ lesz, egyébként 1.

A szimulációs vizsgálatok bizonyították, hogy a *best* egyed alkalmazása 2–5%-al jobb eredményeket biztosít, mint a hagyományos keresztezési eljárás.

3. lépés Mutáció. Három lehetséges mutációs operátort javasoltam:

- Egyszeres mutáció: Ebben az esetben egy véletlenszerűen választott egyed egy véletlen pozíciójában $x_i = 1 - x_i$.
- Kétszeres mutáció: Egy véletlenszerűen kiválasztott egyed két pozícióját az előbbi eset szerint megváltoztatjuk.
- Erős mutáció: Véletlenszerűen kiválasztott egyedben véletlen számú pozíciót változtatunk meg.

Az egyes mutációk alkalmazásának aránya lehet rögzített, de lépésről lépésre adaptívan is változhat a mutációk hatékonyságának függvényében. Megmutattam, hogy kisebb méretű hálózatok esetében a rögzített arányok, míg nagyobb hálózatok esetén az adaptív beállítás biztosítja a jobb végeredményt.

4. lépés Az egyedek költségének meghatározása Az adott RNC szám és elhelyezkedés költséghatékonyságának kiértékeléséhez meg kell határozni az összeköttetéseket is, vagyis az RBS-eket valamilyen módon a megfelelő RNC-hez kell kötni. Algoritmusomban három módszert javasoltam erre a problémára:

1. Kapcsoljunk minden RBS-t a hozzá legközelebb lévő RNC-hez.
2. Használjuk azt a mohó, fa topológia tervező algoritmust, amely a [C1] került publikálásra.
3. Használjuk a 1.2 altézisben javasolt fa topológia tervező algoritmust.

5. lépés Az optimalizálás az 1. lépésnél folytatódik egészen egy meghatározott lépés számig, vagy automatikus leállási kritérium is beállítható, amely meghatározott számú javítás nélküli lépés után leállítja az algoritmust. Szimulációk segítségével megmutattam, hogy az adaptív leállási kritérium alkalmazása 20–25%-al rövidebb futási időt eredményez összehasonlítva a klasszikus genetikai algoritmussal.

1.2. altézis: 2. feladat megoldása – RBS fa topológia tervező algoritmus (RBS Tree Construction Algorithm - TCA)

Javasoltam egy algoritmust, amely képes arra, hogy közel optimálisan meghatározza az összeköttetéseket olyan RBS-ek között, amelyek egy RNC-hez tartoznak.

Jelen eljárás az előző módszer kimenetéből indul ki (RNC-k helye és a hozzájuk kapcsolódó RBS-ek csoportja), és felépíti a hálózatot az RBS-ek és az RNC-k között. Az algoritmust minden RNC-hez tartozó RBS halmazon egymás után le kell futtatni, így kapjuk meg a teljes hozzáférési hálózatot. A javasolt algoritmus a szimulált lehűtés módszeren alapszik, amely széleskörűen használt, bonyolult problémák esetében is jó eredményeket elérő optimalizációs technika. A módszer a fémek lehűlését modellezi, ahol az atomok mindig a minimális összenergiát biztosító állapotba kerülnek (részletesebb leírást találhatunk a [10, 11] cikkekben). Az általam javasolt algoritmus lényege azon RBS-ek számának és helyének változtatása, amelyek közvetlenül kapcsolódnak az RNC-hez (ezeket az RBS-eket a továbbiakban első szintűnek nevezzük és $level_1$ -el jelöljük).

Az optimalizáció főbb lépései a következők:

Kezdeti állapot generálása Néhány RBS véletlenszerű elhelyezése az első szinten.

1. lépés Új állapot a következő lépésekkel érhető el:

- Egy RBS véletlenszerű kiválasztása és első szintre helyezése.
- Egy véletlenszerűen kiválasztott RBS eltávolítása az első szintről.
- Egy első szintű és egy bármely szintű RBS cseréje.

A klasszikus szimulált lehűtésben a fenti alternatívák között rögzített a gyakoriság aránya, de én javasoltam egy olyan stratégiát, amelyben az egyes lépések valószínűsége a sikerességüknek megfelelően változik. Megmutattam, hogy ennek köszönhetően az algoritmus konvergációja gyorsabb és a futási idő csökken.

2. lépés A hálózat előbbi lépés szerinti módosítása után felépítjük a teljes topológiát a [C1] és [C4] cikkekben bemutatott mohó fa építő algoritmus segítségével.

3. lépés Meghatározzuk a költség különbséget a jelenlegi és az új hálózati topológia között. Az alábbi ún. sztochasztikus elfogadási kritérium segítségével a szimulált lehűtés eldönti, hogy az új topológia elfogadható-e vagy sem.

$$P_{accept} = \min\left\{1, \exp\left(-\frac{Cost_{new} - Cost_{orig}}{T}\right)\right\}, \quad (1)$$

ahol $Cost_{new}$ és $Cost_{orig}$ a hálózat teljes költsége a módosítás után és előtt. A T a hőmérsékletnek megfelelő paraméter, amely jelen implementációban exponenciálisan csökken. Ezután az optimalizálás az 1. lépésnél folytatódik.

Befejezés A jobb eredmény elérése érdekében az alábbi lokális javító módszereket alkalmaztam:

- Egy RBS-hez tartozó hálózatrészt egy másik RBS-hez kapcsolunk.
- Két összeköttetést felcserélünk.
- Minden olyan hálózatrészben, amely egy első szintű RBS-hez kapcsolódik meghatározzuk az optimális topológiát.

Különböző hálózatok esetén összehasonlítottam a javasolt fa topológia tervező algoritmusomat a [C1] cikkben javasolt módszerrel. A kapott eredmények az 1. táblázatban láthatók.

Az eredmények mutatják, hogy a TCA átlagosan 8.25%-al alacsonyabb költségű hálózatot képes tervezni, mint a mohó algoritmus. Fontos megjegyezni továbbá, hogy a TCA még nagy hálózatméretek esetén is megfelelően működik. Látható, hogy a TCA eredményeinek szórása kicsi, ami az algoritmus stabilitását bizonyítja.

Az algoritmusok teljesítőképességének vizsgálata

Az általam javasolt módszereken kívül még létezik számos algoritmus, amelyek felhasználhatóak mindkét tervezési probléma megoldására. Nem ismert azonban, hogy a felhasználható módszerek milyen kombinációja biztosítja a legjobb eredményt, ezért kidolgoztam egy olyan keretrendszert, amely lehetővé teszi a különféle algoritmusok tetszőleges kombinációját és ezek összehasonlítását. Az alábbi algoritmusok teljesítőképességét hasonlítottam össze:

1. táblázat: Az általam javasolt és a mohó fa tervező algoritmusok összehasonlítása

#Csomópontok száma	Mohó alg.	TCA	Javítás	TCA szórása
50	10587,3	9340,77	11,8%	0.147%
70	13930,6	12365,4	11,2%	0.383%
100	18252,8	16742,5	8,3%	0.405%
120	21901,9	19883,4	9,2%	0.672%
150	26362	24300,8	7,8%	0.715%
200	33657,4	31434,8	6,6%	1.086%
300	48341,1	45245,4	6,4%	0.972%
500	77144,9	73627	4,6%	1.135%

- 1. módszer: A [C1]-ben publikált Integrált Hálózattervező Algoritmus (Integrated Planning Algorithm - IPA) használata.
 - 2. módszer: Különböző távolság alapú klaszterező módszerek alkalmazása (bővebben [2]-ben) az RNC-k helyének optimalizálására, illetve [C1]-ben javasolt mohó algoritmus az RBS topológia tervezésére.
 - 3. módszer: távolság alapú klaszterező módszerek, majd a TCA alkalmazása.
 - 4. módszer: Az RLA alkalmazása az 1. költség számítási verzióval, majd a TCA használata.
 - 5. módszer: Az RLA alkalmazása a 2. költség számítási verzióval, majd a TCA használata.
 - 6. módszer: Az RLA alkalmazása a 3. költség számítási verzióval.
- Különböző hálózatok esetén meghatároztam a fenti módszerek által szolgáltatott relatív eredményeket (100%-nak tekintetem a legalacsonyabb költségű megoldást, a többi ehhez lett százalékosan viszonyítva). Az eredmények a 2 táblázatban tekinthetők meg. Megmutattam, hogy a legjobb eredmény a 6. verzióval érhető el, de ez a módszer na-

2. táblázat: Különböző tervezési módszerek összehasonlítása

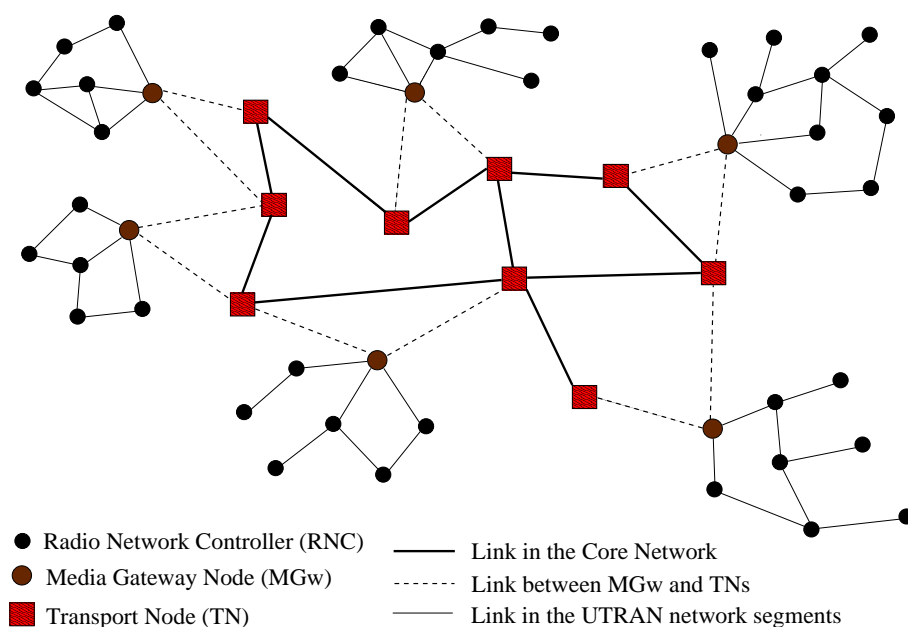
#Csomópontok	1. módszer	2. módszer	3. módszer	4. módszer	5. módszer	6. módszer
100	108.82%	109.76%	107.43%	100.36%	102.54%	100%
200	109.23%	109.92%	106.87%	100.45%	103.34%	100%
300	110.23%	110.54%	107.67%	100.98%	103.45%	100%
500	110.76%	110.57%	107.74%	100.78%	104.12%	100%

gyon idő-igényes, mivel a TCA-nak minden egyes lépésben, minden RNC-hez tartozó klaszterben le kell futnia. Egy jó középút, ha a 4-es módszert használjuk, mivel ennek

futási ideje jóval kisebb az eredmény viszont maximum csak 1%-al lesz rosszabb. Az eredmények alapján kijelenthető, hogy a javasolt algoritmusok segítségével kedvezőbb költségű hálózat tervezhető, mint az ismert módszerekkel.

2. tézis: Költség-alapú UMTS törzs (core) hálózatok tervezése [C9, C10, C13]

Az UMTS Törzs (vagy Gerinc) hálózatának tervezése szintén felvet számos új feladatot, és jelenleg erősen korlátozott azon módszerek száma, amelyek felhasználhatók ezen a területen. Ennek leginkább az az oka, hogy a klasszikus topológia tervezésen kívül itt forgalom elvezetési és méretezési problémák is felvetődnek. Továbbá a nagy hálózati forgalom miatt egy csomópont vagy összeköttetés hibája nagymértékű forgalom veszteséget okoz, ezért a tervezés során hibavédelmi megoldásokra is gondolni kell. Az 1. ábrán az UMTS törzs hálózatának felépítése látható.



1. ábra: UMTS hálózati architektúra

A felmerülő tervezési feladatok az egy UTRAN-hoz tartozó RNC-k összekötése, illetve egy vagy két olyan RNC kiválasztása a hozzáférési és a törzs hálózat közötti összeköttetést biztosító ún. Média Átjárók (Media Gateways - MGw) elhelyezhetők. Továbbá a hozzáférési hálózatokat össze kell kapcsolni egymással, vagyis meg kell tervezni a törzs hálózat Transzport Csomópontjai (Transport Node - TN), valamint a MGw-k közötti kétszeresen összefüggő hálózatot. A topológia tervezésével egyidőben az igények útvonalának meghatározását is meg kell oldanunk. Az összeköttetések és a csomópontok kapacitása és költsége közti összefüggést lépcsős jellegű költség függvény írja le.

Mivel a fenti probléma bizonyítottan \mathcal{NP} -nehéz (visszavezethető az ún. Steiner fa problémára), ezért heurisztikus algoritmusok használata szükséges. Mivel a probléma nagyon nagy, ezért az eredeti feladatot felosztottam két kisebb, jól elkülöníthető alproblémára. Így egy kétfázisú tervezési problémához jutunk, ahol az egyes fázisokban megoldandó feladatokat az alábbiak szerint osztottam fel:

- 1. fázis: E fázisban a feladat a MGw-k helyének, illetve a hálózati topológiának a meghatározása a lépcsős helyett lineáris költség függvény esetén. Mivel a lineáris költség függvény képlete $A_e + B_e * C_e$ jellegű, ezért minden e összeköttetésre meg kell határoznunk A_e (az összeköttetés telepítési költségével arányos) és B_e (a lineáris függvény meredeksége, a lépcsős függvény kapacitás lépcsőiből kerül meghatározásra) értékét. C_e a forgalom összege, amely az e összeköttetésen keresztül halad. Az 1. fázisbeli feladat megoldására egy Szimulált Lehűtés alapú algoritmust javasoltam. (2.1 altézis)
- 2. fázis: Ebben a fázisban már az eredeti lépcsős függvény alapján történik a tervezés. A hálózat topológiáját már nem változtatjuk (kivéve az összeköttetés törlésének lehetőségét), de heurisztikus algoritmusok segítségével az igények egy részének új útvonalat jelölünk ki, ha ez költségcsökkenéshez vezet a lépcsős függvény alapján. (2.2 altézis)

Az eredmények további javítása érdekében javasoltam egy lokális javító módszereket használó utófeldolgozó algoritmust (2.3 altézis).

A probléma felosztása futási idő növekedést okozhat, de a két fázisú optimalizálással jobb eredmények érhetők el, mint az olyan módszerekkel, amelyek az eredeti feladatot oldják meg, mert a jól definiált alproblémák önmagukban sokkal jobban kezelhetők, mint a teljes feladat.

2.1. altézis: 1. fázis megoldása – Algoritmus UMTS törzshálózatok topológiájának tervezésére

Olyan Szimulált Lehűtés alapú algoritmust javasoltam, amely képes megtervezni az UMTS törzshálózatok topológiáját és a forgalmi igények elvezetését közel optimális módon.

Esetünkben a Szimulált Lehűtés algoritmus egy állapota egy hálózati topológia, vagyis az összeköttetések állapota (σ_e jelöli az e összeköttetés állapotát; $\sigma_e = 1$ ha az összeköttetés telepítve van, egyébként 0), illetve a MGw-k helyei. Az állapottér ilyen módon való definiálása biztosítja azt, hogy a globális optimum meghatározható.

Az algoritmus főbb lépései a következők:

Kezdeti lépés A_e és B_e értékek kiszámítása minden összeköttetésre. Adott számú összeköttetést telepítünk, majd az igényeket elvezetjük az összeköttetések hossza szerinti legrövidebb úton.

1. lépés Új hálózati topológia a következő lépésekkel érhető el:

- Összeköttetés telepítés/törlés UTRAN szinten: Egy véletlenszerűen kiválasztott UTRAN hálózatban invertáljuk egy link állapotát ($\sigma_e = 1 - \sigma_e$).
- Összeköttetés telepítés/törlés törzshálózat szinten: Egy törzshálózatbeli összeköttetés (melynek legalább egyik végpontja TN) állapotát invertáljuk.
- MGw mozgatás: Egy véletlenszerűen kiválasztott UTRAN-beli MGw-t egy másik, ugyanabban az UTRAN-ban lévő pontba helyezünk át. Ezek után az összes igénynek, amely az adott UTRAN-ba érkezik vagy onnan indul az új MGw-n keresztül kell kijutnia a törzs hálózatba.

A szimulált lehűtés adaptív módon működik, ami azt jelenti, hogy a különböző lépések (amivel új topológiát lehet elérni) valószínűsége a lépések sikerességétől függ.

2. lépés Bármelyik fenti lépés után a forgalmi igényeket újra elvezetjük Dijkstra algoritmus segítségével, az alábbi súlyfüggvénynek megfelelően, ahol $M \approx (2...5) \cdot \max_e A_e$:

$$w_e = \begin{cases} B_e * C_e & \text{if } \sigma_e = 1 \\ M & \text{if } \sigma_e = 0 \end{cases}$$

3. lépés Az igények elvezetése után kiszámoljuk a hálózat költségét, amely az összeköttetések telepítési (A_e), illetve forgalom függő költségének összege ($B_e * C_e$). Az új hálózati topológia elfogadása vagy elutasítása után az optimalizálás az 1. lépésnél folytatódik tovább a meghatározott maximális iterációs számig.

Alsó korláttal való összehasonlítás megmutatta, hogy a módszerem által eredményezett hálózat költsége és az alsó korlát között átlagosan 11.3% különbség van, ami kedvezőbb, mint a létező azonos feladatot megoldó algoritmusokkal elérhető.

2.2. altézis: 2. fázis megoldása – Két algoritmus az igények elvetésének problémájára UMTS törzshálózatokban

Szimulált lehűtés alapú, valamint általam továbbfejlesztett szimulált allokációs algoritmust javasoltam a felmerülő forgalom elvezetési problémákra 1+1 dedikált védelem esetén.

Mindkét algoritmus az előző feladat végeredményéből indul ki és a forgalmi igényeket újra elvezeti, de már a lépcsős költség függvény szerint. A hálózat topológiája nem változtatható (kivéve az olyan összeköttetések törlését, amelyeken nem megy forgalom), a cél az igények optimális elvezetésének meghatározása.

A SAN algoritmus

A módszer alapja, hogy minden igényhez adott számú lehetséges útvonalat keressünk. Először az igény által használt összeköttetések darabszáma szerinti legrövidebb utakat keressük meg (jelöljük az ilyen út hosszát L_{sp} -vel, majd az $L_{sp} + 1$ hosszú utakat és így tovább. Az eredmények alapján elmondható, hogy az $L_{sp} + 3$ hosszúságú utak már elegendőek a közel optimális megoldás megtalálásához. Bár az előre rögzített utak

szűkítik az állapotteret, de a tesztek bizonyították, hogy $L_{sp} + 3$ -nál hosszabb utakat csak nagyon ritkán használnak az igények, mivel ez már jelentős terhelés növekedést okoz.

Jelöljük a d -edik igényhez tartozó k -adik utat p_k^d , $k = 1 \dots K$, ahol K az igényhez tartozó különböző utak száma. Az optimalizáció egy állapota az, hogy az igények melyik úton kerülnek elvezetésre. Szomszédos állapotba jutunk, ha egy d igényt kiválasztunk és a p_k^d útvonalról a p_l^d útvonalra helyezük át, oly módon, hogy $l \neq k$. A változtatás jóságát a szimulált lehűtés sztohasztikus elfogadási kritériuma segítségével értékeljük ki. Az algoritmus adott számú lépés után befejezi működését.

Mivel az útvonalak előre meghatározottak, ezért az igények elvezetésének sorrendje nem befolyásolja az eredmény jóságát. A módszer további fontos tulajdonsága, hogy az üzemi és a védelmi utakat függetlenül kezeljük az optimalizálás során. A függetlenség azt jelenti, hogy az egy igényhez tartozó üzemi és védelmi utakat nem szükséges egyszerre meghatározni; úgy vannak kezelve, mint egyedi utak, de természetesen az él függetlenség kritérium betartásával.

A továbbfejlesztett SAAL algoritmus

Ez az algoritmus a Michal Pióro által [13]-ben javasolt Szimulált Helyfoglalás (Simulated Allocation – SAAL) módszernek a továbbfejlesztése. A fejlesztéseim az alábbiak voltak:

- olyan adaptív súlyfüggvényt javasoltam, amely elsősorban a hálózat költségének növekedését veszi figyelembe egy adott d igény útvonalának meghatározásánál:

$$w^d(i, j) = \begin{cases} \epsilon + (1 - \frac{c_e^l - C_e + cap^d}{c_e^l}) & \text{ha } C_e + cap^d \leq c_e^l \\ Cost_{link}(C_e + cap^d) - Cost_{link}(C_e) & \text{ha } C_e + cap^d > c_e^l \end{cases}$$

,ahol ϵ egy kis szám, cap^d az elvezetendő igény nagysága, míg c_e^l az összeköttetés jelenlegi kapacitása (az l -edik a lépcső kapacitása a lépcsős költség függvény szerint).

- A klasszikus SAAL algoritmus esetén egy lépésben csak egy igényt vezetünk el vagy törölünk a hálózatból. A javasolt algoritmusomban allokáció esetén néhány kiválasztott igényt egy lépésben vezetünk el. Az igények kiválasztása történhet véletlenszerűen, de csökkenő sorrendbe is rendezhetjük őket kapacitásuk vagy elvezetési költségük szerint. Deallokáció esetén néhány véletlenszerűen kiválasztott igény kerül törlésre a következő lehetőségek szerint:
 - törölünk egy igényt, amely egy véletlenszerűen kiválasztott összeköttetést használ
 - a legterheltebb összeköttetésekről törölünk néhány igényt
 - a legterheltebb összeköttetésről az igények felét töröljük
 - azokat az igényeket töröljük, amelyekkel a legnagyobb költségcsökkenés érhető el

Az ismétlődő igény elvezetések és törlések biztosítják, hogy az állapottér nagy része megvizsgálásra kerül az optimalizálás során. Védelem estén az üzemi és a tartalék utak együtt vannak kezelve (ha egy igényt allokálunk, akkor az üzemi és védelmi útvonala számára is foglalunk helyet).

- A klasszikus SAAL akkor fejezi be működését, ha minden igény elvezetésre került. Javasoltam egy iteratív, többfázisú módszert, amely több klasszikus SAAL optimalizálási ciklusból áll. Minden fázis után adott számú igény törlünk a hálózatból (a túlterhelt élekről, illetve a hálózat alul- illetve túlterhelt részeiről). A törölt igények száma, valamint az igény allokáció valószínűsége fázisról fázisra csökken. Bár ez az eljárás sokkal időigényesebb, mint a klasszikus SAAL, de 4-6%-al jobb eredmények érhetőek el vele.
- A klasszikus SAAL esetén az igény allokáció valószínűsége változatlan az optimalizáció során. Az alábbi módszert javasoltam a valószínűség adaptív változtatására:
 - amikor az igény elvezetések sikeresek, az allokáció valószínűsége nő a gyorsabb konvergáció érdekében
 - ha néhány igény elvezetés sikertelen volt (nagyobb kapacitású összeköttetés használatára van szükség) a valószínűség csökken, hogy könnyebben el lehessen hagyni a jelenlegi hálózati állapototA fenti adaptív valószínűség beállítást alkalmazva 2-3% költség csökkenés érhető el a klasszikus SAAL módszerhez képest.

2.3. altézis: Utófeldolgozó algoritmus az eredmények további javítására

Néhány lokálisan javító módszert javasoltam, amelyek kedvezőbb költségű hálózati konfigurációt képesek elérni az igények útvonalának módosításával.

A motiváció az volt, hogy bár az előzőekben vázolt algoritmus képes jó minőségű eredmények elérésére, de a legtöbb esetben lokális eljárások segítségével az eredmény tovább javítható. Ennek érdekében az alábbi négy hatékony módszert javasoltam:

1. E módszer alapötlete az, hogy ha az alacsony kihasználtságú összeköttetésekről eltávolítunk és másfelé irányítunk igényeket, akkor ezen összeköttetések költsége csökkenthető. Olyan összeköttetéseket választunk ki, amelyek relatív terheltsége kicsi, és annyi igényt távolítunk el ezekről, amennyi ahhoz szükséges, hogy egy kisebb kapacitású (és így olcsóbb) összeköttetést tudjunk használni. Majd a törölt igények újra elvezetésre kerülnek egy speciális költségfüggvény segítségével, olyan utakat biztosítva az igényeknek, amelyek minimális költségnövekedést eredményeznek.
2. Sok esetben az igények elvezetése nem optimális. Ez az eljárás képes ezt ellenőrizni, illetve egyenként újra elvezetni bizonyos igényeket, ha ezzel költség csökkenés

érhető el. Jelen módszer kiszűri a szükségtelenül hosszú utakat, valamint javítja a hálózat terhelés-egyensúlyát.

3. Ha egy összeköttetés kapacitását megnöveljük, akkor ez néhány más összeköttetés kapacitásának csökkenését eredményezheti, amely összességében költség csökkenést is jelenthet. Ennek érdekében egyenként, minden összeköttetés kapacitását megnöveljük (pl. 155Mbps összeköttetés helyett 622 Mbps-t használhatunk), járulékos költség nélkül. Az igényeket elvezetjük, majd a megnövelt kapacitású összeköttetés járulékos költsége is a hálózat költségéhez adódik. Ez az eljárás elősegíti, hogy a hálózat "központjában" lévő összeköttetések – amelyek a forgalom nagy részét szállítják – elegendően nagy kapacitásúak legyenek.
4. A hálózat költsége drasztikusan csökkenthető ha a nélkülözhető összeköttetéseket meg sem építjük (vagy nem béreljük). Jelen módszer megvizsgálja az összeköttetéseket és a nélkülözhetőket törlésre kerülnek.

Megjegyzendő, hogy bár az Utófeldolgozó algoritmust már előzőleg optimalizált hálózatokon való működésre dolgoztam ki, de használható "zöld-mezős" tervezési feladatok megoldására is; ebben az esetben az eredmény messze lesz az optimumtól.

Szimulációk segítségével megmutattam, hogy az Utófeldolgozó algoritmus képes a hálózatok költségét 6–18%-al csökkenteni (a bemeneti hálózat előzetes tervezésétől függően).

3. tézis: Adminisztratív súlyok optimalizálása OSPF hálózatokban [J5, C2, C3, C11]

Az OSPF (Open Shortest Path First) a legáltalánosabban használt útvonalválasztó protokoll az Internet Protokoll (IP) használó hálózatokban. Az OSPF az összeköttetésekhez rendelt adminisztratív súlyok szerinti legrövidebb úton továbbítja a csomagokat, illetve ha több azonos súlyú legrövidebb út van, akkor alkalmazza az ún. Azonos Súlyú Többszörös-Utak (Equal-Cost MultiPath – ECMP) szabályt. Az OSPF hálózatokban felmerülő optimalizálási probléma a súlyok olyan beállítása, amely a legjobb hálózat teljesítőképességet eredményez. Mivel a kutatási terület újnak mondható, még kevés eredmény született ezen a téren: [?],[20],[21] a legfőbb irányadó művek. A probléma amellyel foglalkozom a folyamproblémák témakörébe sorolható, kiegészítve azzal a megkövetéssel, hogy figyelembe kell venni az OSPF útvonalválasztási szabályokat is. Szerzőtársaim a [C3] cikkben bizonyítják, hogy a probléma \mathcal{NP} -teljes, tehát heurisztikus algoritmus használatára van szükség a megfelelő eredmény elfogadható időn belüli eléréséhez. A feladat megoldása során feltételeztem, hogy az adminisztratív súlyok nem változnak dinamikusan, hanem rövid időintervallumon belül állandónak tekinthetők. A probléma megoldására két algoritmust javasoltam.

3.1. altézis: Mohó algoritmuson alapuló egyszerű súlybeállítás (Simple Weight Adjustment – SWA)

Gyors, hatékony módszert javasoltam, amely az OSPF súlyokat rövid futási idő alatt, elfogadható eredménnyel képes beállítani.

E módszer kifejlesztésénél a legfontosabb tényező az volt, hogy a *lehető legrövidebb futási idő alatt* használható, jó minőségű megoldáshoz juthassunk. Ez lehetővé teszi, hogy a módszert olyan helyzetekben alkalmazzuk, ahol ún. valós-idejű (on-line) súlybeállítási problémákkal kerülünk szembe (pl. összeköttetés hibák vagy drasztikus forgalom ingadozással járó szituációk) ahol a gyors újra-konfigurálás nagyon fontos. Az algoritmus az ún. *egyfázisú* súlybeállítási megközelítésen alapszik, ami azt jelenti, hogy a súlyokat iteratív eljárás keretében, közvetlenül számítjuk ki a hálózat kihasználtsága és egyéb paraméterek alapján. Jelen algoritmus képes valós és egész számú súlyokat is kezelni.

A javasolt módszer a két alábbi lokális kereső algoritmusból áll:

– Súly beállítás (Weight Adjustment – WA): A módszert akkor alkalmazzuk, ha a hálózat túlterhelt (legalább egy összeköttetés túlterhelt). A túlterhelt összeköttetések súlyát a túlterhelés jelenlegi és néhány előző iterációbeli értékének függvényében növeljük. Mivel több előző lépésbeli terhelésértéket veszünk figyelembe, ezért a ciklusok (egymás utáni súlynövelések és csökkentések) megelőzhetők, továbbá a konvergáció is egyenletesebb lesz. A szabad kapacitással rendelkező összeköttetések súlya a rendelkezésre álló kapacitás függvényében csökkentve lesz. Célunk egy olyan súlyrendszer megtalálása, amelynél a hálózatban nincs túlterhelt összeköttetés.

– Terhelés kiegyenlítés (Load Optimization – LO): Ha a hálózatban nincs túlterhelés, akkor bizonyos összeköttetések súlyát kismértékben megváltoztatjuk a terhelés kiegyenlítése érdekében. Jelen szakaszban célunk a szabad kapacitás értékének maximalizálása.

Ha az optimalizálás során nem találunk megvalósítható megoldást (vagyis lesz legalább egy túlterhelt összeköttetés a hálózatban), akkor két kimenet közül lehet választani: a túlterhelt összeköttetés száma vagy a túlterhelés értéke legyen minimális. Ha az algoritmus talál olyan súlybeállítás(oka)t ahol nincs túlterhelés, akkor a kimenet az a súlyrendszer lesz, amely a legjobb hálózat kihasználtságot eredményezi.

Szimulációk segítségével megmutattam, hogy az SWA képes a létező módszerek megoldásaival összemérhető eredményeket produkálni, de jóval rövidebb (a legtöbb esetben egy nagyságrenddel kisebb) futási idő alatt.

3.2. altézis: A súly optimalizáló algoritmus (Weight Optimizer – WO)

*Hatékony, robusztus és skálázható **determinisztikus** módszert javasoltam OSPF súly beállításra, amely jobb eredményt képes elérni, mint a létező algoritmusok.*

Jelen módszer kidolgozásánál az elsődleges célok a robusztusság, a skálázhatóság (a futási idő és az eredmény minősége között), a determinisztikus működés, valamint

az ún. *fekete doboz (black box)* típusú költségfüggvény alkalmazása voltak. A módszer legfontosabb tulajdonsága a determinisztikus működés, ami biztosítja, hogy egy futtatás elegendő a lehető legjobb eredmény megtalálásához. A WO módszer szintén az *egyfázisú* súlybeállítási megközelítésen alapszik.

Az optimalizálás során a hálózat két állapotban lehet:

- Túlterhelt állapot (Overloaded state – OLS): Ebben az állapotban legalább egy összeköttetés túlterhelt.
- Normál állapot (Normal state – NS): Ebben az állapotban minden összeköttetés terheltsége kisebb, mint a rendelkezésre álló kapacitás.

Túlterhelt állapotban az algoritmus két eljárást használ, amelyek megpróbálnak egy túlterhelés mentes hálózati állapotba eljutni, más szavakkal ebben a szakaszban egy olyan állapotot keresünk, amely megfelelő kiindulópont lehet a további (normál állapotbeli) optimalizálás számára. Ebben a fázisban az algoritmus egy módosított költség függvényt használ, amely a túlterhelt linkek számától, valamint a hálózat túlterhelésének statisztikai paramétereitől (átlagos túlterhelés értéke, illetve a terhelés eloszlása a hálózatban) függ. Túlterhelt állapotban az algoritmus az alábbi két eljárást használja:

- Az igények ismételt elvezetése segítségével az egyik eljárás a tehermentesíti a túlterhelt összeköttetéseket, valamint kiegyenlíti a hálózat terheltségét.
- A másik eljárás megszünteti a jelentősen túlterhelt hálózat részeket (ahol több túlterhelt összeköttetés kapcsolódik egy közös csomóponthoz).

Ha az algoritmus talált egy olyan súlyrendszert, amely normál hálózati állapotot eredményez, akkor a továbbiakban a felhasználó által definiált vagy az előre rögzített (szabad kapacitás maximalizálása) célfüggvényt használja. Ebben a fázisban az alábbi lokális kereső módszereket használtam:

- A legterheltebb és a legtöbb szabad kapacitással rendelkező összeköttetés súlyának módosítása a kihasználtságuk alapján.
- Ez az eljárás az összeköttetések relatív terheltségének szórását próbálja csökkenteni a jobb hálózat kihasználtság elérése érdekében.
- Ez az eljárás a terhelést egyenlíti ki a különböző hálózatrészek között.

Sok esetben a normál állapotban végrehajtott módosítások ismét túlterhelés(ek)e)t okozhatnak, ezért az algoritmus iteratív módon működik a két hálózati állapot között. Ez a működési mód nagyban megkönnyíti az optimalizálás során elért lokális optimum pontok elhagyását. Az algoritmus befejezi működését, ha adott lépésszám alatt egyik normál állapotbeli eljárás sem képes már javítani (a lépésszámot az algoritmus állítja be a hálózat mérete alapján).

Szimulációk segítségével megmutattam, hogy a WO módszer képes számottevően jobb eredményeket elérni, mint az ismert módszerek, továbbá a determinisztikus működés biztosítja a stabilitást.

Az algoritmusok teljesítőképességének vizsgálata

A javasolt módszereimet összehasonlítottam ismert heurisztikákkal és az eredmények kivonatát a 3. táblázatban összegeztem. Az összehasonlításokhoz két algoritmust használtam: az egyik egy szimulált lehűtés alapú algoritmus (SAN), míg a másik módszer a szimulált allokációt (SAL) használja. Az algoritmusokról további információ található a [J5] cikkben. A táblázatban használt jelölések a következők: AVRT - átlagos futási idő, FNCR - a szabad kapacitás aránya százalékban. A bemutatott eredmények 5 és 10 százalékkal túlméretezett 28 pontos hálózaton készültek.

3. táblázat: Létező algoritmusok és a javasolt módszerek összehasonlítása

Net28_5%	WO	SWA	SAN	SAL
AVRT	68.29s	19.36s	~15min	47.38s
FNCR	5.00	4.9	4.8	4.9
Net28_10%	WO	SWA	SAN	SAL
AVRT	80.8s	11.43s	~15min	16.95s
FNCR	9.5	9.2	9.4	9.2

Megmutattam, hogy a WO algoritmus hozzávetőlegesen 3% – 6%–al jobb eredményt ad, mint a korábban publikált módszerek, míg az SWA módszer a leggyorsabb és az általa szolgáltatott eredmények elég kedvezőek. A javasolt algoritmusok teljesítőképességét 7–120 csomópontig terjedő hálózatokon vizsgáltam és az eredmények hasonlóak voltak a fentiekhez.

4. tézis: Dinamikus útvonalválasztás és hullámhossz foglalás többszálás WDM optikai hálózatokban [J7]

A WDM technikának köszönhetően az optikai hálózatok jelentősen nagyobb adatmennyiség továbbítására képesek, mint más adatátviteli módok. Eddig a WDM hálózatok általában ún. statikus módban működtek, ami azt jelenti, hogy az igényekhez előre meghatározott hullámhosszak és utak tartoztak. Mivel az egyre nagyobb súlyt kapó adat forgalom borsztös jellegű, ezért az átvinni kívánt forgalmi igények inkább dinamikus modellt igényelnek. Mivel tehát adatforgalom esetén a dinamikus erőforrás konfigurálás jobb teljesítőképességet eredményez, ezért ez a technika előtérbe került az utóbbi időben. A megjelenő optimalizálási feladatok a következők: olyan (a) megfelelő hullámhossz kiválasztása (wavelength selection [WS]), valamint (b) alkalmas útvonal kiválasztása (wavelength routing [WR]) a beérkező igény számára, amely a lehető legkedvezőbb hosszútávú blokkolási valószínűséget biztosítja. Számos algoritmust javasoltak már erre a problémára, de egyrészt ezek nem használtak adaptív útvonalválasztást, másrészt a két fenti részfeladatot külön kezelték, amely számottevően rontotta az eredmények jóságát. Úgy találtam, hogy a WS és a WR problémák együtt való kezelése,

illetve az adaptív (hálózat állapot függő) útvonal választás alkalmazása jobb eredményeket biztosít, mint az ismert algoritmusok.

4.1. altézis: Egy (on-line) algoritmus dinamikus erőforrás foglalásra WDM hálózatokban

Javasoltam egy algoritmust (WS&WR), amely a hullámhossz kiválasztás (WS) és az útvonal kiválasztás (WR) problémáját a hálózat jelenlegi állapotát felhasználva integrált módon oldja meg, így jobb eredményeket szolgáltatva, mint az ismert módszerek.

Feltételeztem, hogy a vizsgált hálózatban nincs hullámhossz konverzió (a konverterek magas árával indokolva), valamint egy összeköttetés tartalmazhat több üvegszálat is. Az alábbiakban összefoglaltam a további fontosabb feltételeket:

- A hálózati topológia és a rendelkezésre álló erőforrások (összeköttetésenkénti üvegszálak és üvegszálankénti hullámhosszak száma) ismert és nem változik az algoritmus működése alatt (kivéve az esetleges hibákat, amelyeket az algoritmus képes kezelni).
- Minden beérkező igény forrás és cél pontja ismert, de az igény tartási ideje nem.
- Nem rendelkezünk előzetes információval a jövőben beérkező igényekről, vagyis az algoritmus csak a hálózat aktuális állapotát veheti figyelembe működése során.
- Feltesszük, hogy egy igény csak egy hullámhosszat foglalhat.

Az általam javasolt algoritmus a következő főbb lépésekből áll:

1. lépés Új igény foglalási kérés érkezik a hálózatba.

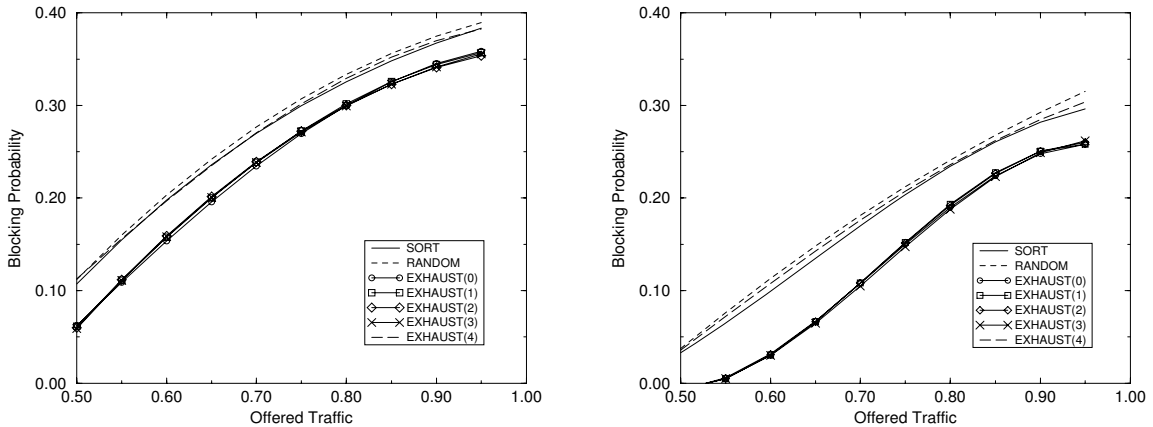
2. lépés Az általam javasolt integrált hullámhossz és útvonal választási stratégiával minden egyes hullámhosszon megpróbáljuk elvezetni az adott igényt.

- A hullámhossz választási (WS) feladat esetén 5 eljárást javasoltam arra hogyan tudjuk mérni egy adott hullámhossz alkalmasságát az igény szállításra. A legfontosabb tényezők az igény útjának hossza valamint a hullámhossz terheltsége.

- Az útvonal választási (WR) feladat megoldására néhány súlyfüggvényt javasoltam, amelyek segítségével az érkező igények elvezethetők egy adott hullámhosszon. A súlyfüggvények adaptívak; különféleképpen veszik figyelembe az összeköttetések terheltségét.

- A WS&WR algoritmusom figyelembe veszi a különböző igények közötti esélyegyenlőség megtartását, biztosítva azt, hogy a különböző felhasználók által észlelt egyéni igény blokkolási valószínűség és a hálózatbeli átlagos blokkolási valószínűség közötti különbség minimális legyen.

3. lépés Ha lehetséges akkor a beérkező igényt elvezetjük a hálózatban, egyébként az igény blokkolódik.



2. ábra: Blokkolási valószínűség különböző WS stratégiák esetén a Net_{25} és Net_{56} hálózatokban

A 2. ábrán az általam javasolt WS módszereket hasonlítottam össze az ún. Első Alkalmos Hullámhossz (First-Fit), valamint a Véletlen Hullámhossz Választással, 25 és 56 pontos hálózatok esetén.

Az eredmények alapján kijelenthető, hogy 4 javasolt WS eljárás (amelyek figyelembe veszik az adott hullámhossz terheltségét, valamint az igény útvonalának hosszát és az ebből adódó terhelés növekedést) 5–10%-al alacsonyabb blokkolási valószínűséget biztosít, mint az eddig javasolt algoritmusok bármely útvonal választáshoz használt súlyfüggvény, hálózatméret, valamint igény beérkezési intenzitás esetén. Az 5. WS stratégia (amely csak az útvonal terhelését veszi figyelembe) a létező módszerekkel összemérhető eredményt biztosít.

4.2. altézis: WDM hálózatok méretezése és konfigurálása

Az előző altézisben ismertetett algoritmus egy lehetséges alkalmazását javasoltam a WDM hálózatokban felmerülő méretezési és konfigurálási feladatok oly módon való megoldására, amely jól illeszkedik a dinamikus erőforrás foglaláshoz.

Feladatunk egy előírt blokkolási valószínűség (P_{goal}) eléréséhez szükséges üvegszálak összeköttetésenkénti ($F_{num}(l)$, $l \in \mathcal{L}$, ahol \mathcal{L} az összeköttetések halmazát jelöli) és a hálózatban használt hullámhosszak (W_{num}) számának meghatározása ismert igény beérkezési folyamat esetén.

Az általam javasolt módszer főbb lépései az alábbiak:

Kezdeti lépés Állítsuk W_{num} és $F_{num}(l)$ értékét 1-re.

1. lépés Az adott hálózati erőforrásokat (üvegszálak és hullámhosszak száma) használva az előző tézisben ismertett erőforrás fogláló algoritmussal megpróbáljuk az érkező

igényeket elvezetni a hálózatban. Így egy működő hálózatot szimulálunk, ahol mérjük az elérhető blokkolási valószínűséget (P_{curr}). Kellő számú igény beérkezése után megkapjuk P_{curr} stabil állapotbeli értékét, és ezt összehasonlítjuk P_{goal} -al. Ha $P_{goal} < P_{curr}$ akkor a hálózat jelenlegi erőforrásai nem elegendők a kitűzött blokkolási valószínűség eléréséhez, további hálózatbővítésre van szükség. Ha $P_{goal} \geq P_{curr}$ az optimalizálás befejeződik.

2. lépés – *Hálózatbővítés* A hálózat bővíthető ha $F_{num}(l)$, W_{num} vagy mindkettő értékét megnöveljük az alábbi módokon:

- $F_{num}(l) = F_{num}(l) + n(l)$ minden $l \in \mathcal{L}$ összeköttetésre
- $W_{num} = W_{num} + m$
- $F_{num}(l) = F_{num}(l) + n(l)$ minden $l \in \mathcal{L}$ összeköttetésre és $W_{num} = W_{num} + m$

ahol $n(l)$ és m egész számok, amelyeket vagy a felhasználó állít be, vagy az algoritmus határoz meg lépésről lépésre a P_{goal} és P_{curr} közötti különbség alapján ($n(l)$ és m értéke csökken az előbbi különbség csökkenésével). A hálózat bővítése után az optimalizálás az 1. lépésnél folytatódik.

3. lépés – *Befejezés* Lehetséges, hogy az eredményként kapott hálózat a szükségesnél nagyobb kapacitással rendelkezik (ez akkor lehetséges, ha $n(l)$ vagy m értéke az optimalizálás során nagyobb volt mint 1). Ezért utolsó lépésként az algoritmus a hálózat erőforrásait egyesével csökkenti, miközben természetesen P_{curr} -t folyamatosan ellenőrzi. Ily módon a hálózat nagyon pontosan méretezhető adott blokkolási valószínűség és igény beérkezési folyamat esetén. Az optimalizálás eredményeként azt a konfigurációt kapjuk meg, amelynél P_{curr} még éppen kisebb volt, mint az előírt érték.

A fenti módszer használható a statikus méretezési feladat megoldására is az alábbi módon. Az algoritmus rögzített számú igénnyel dolgozik (amennyit el kell vezetni a hálózatban); P_{goal} értékét 0-ra állítjuk, ami azt jelenti, hogy olyan hálózati állapotot keresünk amelynél minden igény egyidőben el van vezetve. Az ily módon kapott állapot egy lehetséges statikus konfiguráció. Az algoritmus ezek után még adott számú hálózati konfigurációt keres, amelyek közül a legkevesebb erőforrást használó lesz kiválasztva (amely a legtöbb szabad erőforrást tartalmazza ugyanazon $F_{num}(l)$ és W_{num} értékek esetén).

Különböző igényszám és eloszlás esetén 50 véletlenszerűen generált teszhálózaton összehasonlítottam az algoritmusomat a [35] cikkben publikált módszerrel és az alábbi eredményeket kaptam:

- A javasolt módszerem képes volt 8%-al kevesebb erőforrást használva ugyanannyi igényt elvezetni.
- Ugyanazon $F_{num}(l)$ and W_{num} értéknél algoritmusom képes volt olyan útvonal elvezetést találni, amely 5%-al több szabad kapacitást biztosít.

5. Az Eredmények alkalmazhatósága

Az 1. téziscsoportban ismertetett algoritmusok az Ericsson Kft UMTS hálózattervező és analízáló eszközében kerültek felhasználásra. A 2. téziscsoportban bemutatott algoritmusok egy része az Ericsson UMTS gerinchálózat tervező eszközébe lesz beépítve. Az OSPF adminisztratív súly optimalizálással kapcsolatos eredmények az Ericsson magyarországi kutatólaboratóriuma valamint a Varsói Műszaki Egyetem együttműködéséből születtek, és ezeket mindkét intézmény felhasználta. A WDM hálózatokban felmerülő dinamikus útvonal elvezetés problémáját megoldó algoritmus az Ericsson Magyarországi Kutatólaboratóriuma valamint a Budapesti Műszaki és Gazdaságtudományi Egyetemen működő Nagysebességű Hálózatok Laboratóriuma közti együttműködés keretében került kidolgozásra, és várhatóan egy WDM hálózattervező eszközben kerül felhasználásra.

6. Köszönetnyilvánítás

Szeretném megköszönni konzulensem Dr. Sallai Gyula szakmai vezetését.

A folyamatos támogatásért hálás vagyok Dr. Boda Miklósnak, az Ericsson Magyarországi Kutatás és Fejlesztés Központja vezetőjének, és Dr. Hans Erikssonnak, az Ericsson Traffic Analysis and Network Performance Laboratóriuma vezetőjének. Szintén köszönetemet fejezem ki Dr. Henk Tamásnak, a Nagysebességű Hálózatok Laboratóriuma vezetőjének.

Szeretném köszönetemet kifejezni kollégámnak, Dr. Szentesi Áronnak értékes és folyamatos segítségéért, valamint hasznos ötleteiért és tanácsaiért.

Külön köszönetet szeretnék mondani az Ericsson Traffic Laboratóriumban és a Nagysebességű Hálózatok laboratóriumban dolgozó kollégáimnak: Jüttner Alpárnak, Gódor Istvánnak, Dr. Magyar Gábornak, Weiner Zsuzsának, Dr. Laborczi Péternek, Dékány Tamásnak, Szviatovszki Balázsnak, Szlovensák Attilának, Orincsay Dánielnek, Józsa Balázsnak, Dr. Cinkler Tibornak, valamint Veres Andrásnak.

Az eredményes együttműködésért és a 2. valamint 3. tézisekben nyújtott segítségért hálás vagyok Piotr Gajowniczek-nek, Andrzej Myslek-nek, Stanislaw Kozdrowski-nak, valamint Michal Pióro-nak.

Különösen szeretném megköszönni szüleim és testvérem folyamatos támogatását.

Szeretnék köszönetet mondani azoknak, akik barátként segítettek és támogattak: Baranyai Péternek, Gödör Áginak, Gödör Istvánnak, Háda Bálintnak, Maricza Istvánnak, Rimányi Istvánnak, Suhai Györgynek, Szászik Mónikának, Simovits Móninak, Szlávik Árpának, valamint Molnár Sándornak.

Végezetül szeretnék mindent megköszönni kedvesemnek, *Sólyom Eszternek*.

Irodalom

- [1] T. Ojanperä, R. Prasad, *Wideband CDMA for Third Generation Mobile Communication*, 1998, Artech House Publishers.
- [2] Z. Drezner (editor), *Facility Location*, Springer Series in Operations Research, 1995
- [3] R. Metwani, P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995
- [4] P. M. Camerini, G. Galbiati, F. Maffioli, *Complexity of Spanning Tree Problems*, European Journal of Operational Research, 1980.
- [5] P. M. Camerini, G. Galbiati, F. Maffioli, *On the Complexity of Finding Multi-constrained Spanning Trees*, Discrete Applied Mathematics vol.5, 1983
- [6] S. C. Narula, C. A. Ho, *Degree-constrained minimum spanning tree*, Comput. Ops. Res. 7. 1980.
- [7] B. Boldon, N. Deo, N. Kumar, *Minimum-weight Degree-Constrained Spanning Tree Problem: Heuristic and Implementation on an SIMD parallel machine*, Technical Report, 1995, Dep. of Computer Science, University of Central Florida, Orlando
- [8] S. Dravida, H. Jiang, M. Kodialam, B. Samadi, Y. Wang, *Narrowband and Broadband Infrastructure Design for Wireless Networks* IEEE Communications Magazine, May, 1998.
- [9] P. Kallenberg, *Optimization of the Fixed Part GSM Networks Using Simulated Annealing*, Proc. Networks 98, 8th International Telecommunication Network Planning Symposium, Sorrento, Italy, October 1998.
- [10] B. Hajek, *A Tutorial Survey of Theory and Applications of Simulated Annealing*, Proc. 24th Conference on Decision and Control, Ft. Lauderdale, FL., December 1985.
- [11] B. Hajek, *Cooling Schedules for Optimal Annealing*, Mathematics of Operation Research, Vol 13, No.2, May 1988
- [12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wiley, 1989
- [13] P. Gajowniczek, M. Pióro, A. Arvidsson, *VP reconfiguration through Simulated Allocation*, NTS 13, Thirteenth Nordic Teletraffic Seminar, August 1996.
- [14] B. Fortz, M. Labbé, F. Maffioli, *Methods for Designing Reliable Networks with Bounded Meshes*, 15th ITC (International Teletraffic Congress), 1997.
- [15] M. Zethzon, T. Cinkler, I. Andersson, *Greedy Algorithms for Topological Design* Proc. Networks 98, 8th International Telecommunication Network Planning Symposium, Sorrento, Italy, October 1998.
- [16] K. Holmberg, J. HellStarnd, *Solving the Uncapacitated Network Design Problem by a Lagrangean Heuristic and Barnch and Bound* Operations Research 46: 247-259, 1998.
- [17] *OSPF Version 2*. RFC2328.txt, www.ietf.org/rfc/rfc2328.txt, 1998
- [18] B. Fortz, M. Thorup, *Internet Traffic Engineering by Optimizing OSPF Weights*, Proc. INFOCOM 2000, Tel-Aviv, 2000
- [19] W. B. Ameur, E. Gourdin, B. Liau *Internet Routing and Topology Problems*, Proc. DRCN 2000, Design of Reliable Communication Networks, Munich, 2000
- [20] W. B. Ameur, E. Gourdin, B. Liau *Dimensioning of Internet Networks*, Proc. DRCN 2000, Design of Reliable Communication Networks, Munich, 2000
- [21] A. Faragó, B. Szviatovszki, Á Szentesi, *Allocation of Administrative Weights in PNNI*, Proc. Networks 98, 8th International Telecommunication Network Planning Symposium, Sorrento, Italy, October 1998.
- [22] G. Kotelly, *Worldwide fiber-optic markets to expand unabated*, Lightwave, Vol. 13, No. 13, pp. 6-8, December 1996
- [23] I. Chlamtac, A. Ganz, G. Karmi, *Lightpath communications: an approach to high bandwidth optical WAN's*, IEEE Transactions on Communications, Vol. 40, no. 7, 1992
- [24] S. Xu, L. Li, S. Wang, *Dynamic Routing and Assignment of Wavelength Algorithms in Multifiber*

- Wavelength Division Multiplexing Networks*, IEEE Journal on Selected Areas in Communications, Vol. 18 No. 10, October 2000
- [25] X. Zang, C. Qiao, *Wavelength Assignment for Dynamic Traffic in Multi-fiber WDM Networks*, ICCCN'98, Vol. 18 No. 2, Lafayette, LA, 1998
 - [26] E. Karasan, E. Ayanoglu, *Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks*, Proc. IEEE GLOBECOM, London, England, November 1996
 - [27] A. Mokhtar, M. Azizoglu, *Adaptive wavelength routing in all-optical networks*, submitted to IEEE/ACM Transactions on Networking
 - [28] S. Subramaniam, R. Barry, *Wavelength assignment in fixed routing WDM networks*, Proc. IEEE ICC, Montreal, Canada, November 1997
 - [29] A. Birman *Computing approximate blocking probabilities for a class of all-optical networks*, Proc. IEEE INFOCOM, April 1995
 - [30] H. Harai, M. Murata, H. Miyahara, *Performance Of alternate routing methods in all-optical switching networks*, Proc. INFOCOM, April 1997
 - [31] L. Li, A. K. Somani, *Dynamic Wavelength Routing Using Congestion and Neighborhood Information*, IEEE/ACM Transactions on Networking, Vol. 7, No. 5, October 1999
 - [32] Nilesh M. Bhide, Krishna M. Sivalingam, *Routing mechanisms employing adaptive weight functions for shortest path routing in optical WDM networks*, Photonic Network Communications, Vol. 3. No. 3. 2001
 - [33] E. Karasan, S. Banerjee, *Performance of WDM Transport Networks*, IEEE Journal on Selected Areas in Communications, Vol. 16, p. 764-779, September 1998
 - [34] E. Karasan, E. Ayanoglu, *Performance comparison of reconfigurable wavelength selective and wavelength interchanging cross-connect in WDM transport networks*, Proc. 3rd IEEE/COMSOC Workshop on WDM Network Management and Control, Montreal, Canada, November 1997
 - [35] G. Conte, M. Listanti, R. Sabella, M. Settembre *Strategy for protection and restoration of optical paths in WDM backbone networks for next generation Internet infrastructures*, accepted for publication on Journal of Lighthwave Technology, 2002

Publikációk

Folyóiratcikkek

- [J1] J. Harmatos *Számlázás ATM rendszerben*, Magyar Távközlés, 9. évfolyam, 5. szám, 1998 Május, pp. 21–26
- [J2] J. Harmatos *A hálózatok fizikai rétegének méretezése*, Magyar Távközlés, 9. évfolyam, 11. szám, 1998 November, pp. 12–15
- [J3] G. Bóné, J. Harmatos *Algorithms in Network Planning and Management*, Hungarian Telecommunications Periodical, Selected Papers of 1999, 1999, pp. 33–38
- [J4] G. István, J. Harmatos, *Az UMTS hozzáférési hálózatának tervezése*, Magyar Távközlés, 2000 Augusztus
- [J5] M. Piòro, Á. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, S. Kozdrowski: *On OSPF Related Network Optimisation Problems*, Performance Evaluation, Vol 48, Nr 1–4, May 2002, pp. 201–223
- [J6] A. Szlovencsák, I. Gódor, J. Harmatos, T. Cinkler: *Planning Reliable UMTS Terrestrial Access Networks*, IEEE Communications Magazine, January 2002, pp. 66–72
- [J7] J. Harmatos, P. Laborcz: *Dynamic Routing and Wavelength Assignment in Survivable WDM Networks*, Photonic Network Communications, Vol. 4, Nr. 3–4, July–December 2002, pp. 357–375

Konferencia-előadások

- [C1] J. Harmatos, A. Jüttner, Á. Szentesi: *Cost-based UMTS Transport Network Topology Optimization*, International Conference on Computer Communication, ICCC'99, Tokyo, Japan, September 1999, pp. 00111–1–8
- [C2] M. Piòro, Á. Szentesi, J. Harmatos, A. Jüttner, S. Kozdrowski: *On OSPF Related Networks Optimization Problems*, 8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks, Ilkley, UK, July 17–19, 2000, pp. 70/1–70/14
- [C3] A. Jüttner, J. Harmatos, Á. Szentesi, M. Piòro: *On Solvability of an OSPF Routing Problem*, 15th Nordic Teletraffic Seminar, Lund, Sweden, August 22–24, 2000, pp. 1–9
- [C4] J. Harmatos, Á. Szentesi, I. Gódor: *Planning of Tree-Topology UMTS Terrestrial Access Networks*, 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2000, London, UK, September 18–21, 2000, pp. 353–357
- [C5] P. Gajowniczek, M. Piòro, Á. Szentesi, J. Harmatos: *Solving an OSPF Routing Problem with Simulated Allocation*, First Polish–German Teletraffic Symposium, PGTS 2000, Dresden, Germany, September 24–26, 2000, pp. 177–184
- [C6] A. Jüttner, J. Harmatos, D. Orincsay, B. Szviatovszki, Á. Szentesi: *On-demand Optimization of Label Switched Paths in MPLS Networks*, In proc. of the 9th International Conference on Computer Communications and Networks, IEEE ICCCN 2000, Las Vegas, USA, October 16–18, 2000, pp. 107–113
- [C7] D. Orincsay, J. Harmatos: *On-demand Optimization of Protected LSP Tunnels in MPLS Networks*, 9th IFIP Conference on Performance Modelling and Evaluation of High Speed Networks, Budapest, Hungary, June 27–29, 2001, pp. 357–368
- [C8] A. Szlovencsák, I. Gódor, T. Cinkler, J. Harmatos: *Planning Reliable UMTS Terrestrial Access Networks*, 3th International Workshop on Design of Reliable Communication Networks, DRCN 2001, Budapest, Hungary, October, 2001, pp. 84–90
- [C9] M. Piòro, A. Jüttner, J. Harmatos, Á. Szentesi, P. Gajowniczek, A. Myslek: *Topological Design of Telecommunication Networks*, 17th International Teletraffic Congress, ITC 17, Salvador da Bahia, Brazil, December 2–7, 2001, pp. 629–642
- [C10] M. Piòro, A. Jüttner, J. Harmatos, Á. Szentesi, A. Myslek: *Topological Design of MPLS Networks*, GLOBECOM 2001, San Antonio, USA, 25–29 November, 2001, pp. 12–16
- [C11] J. Harmatos: *A Heuristic Algorithm for Solving the Static Weight Optimization Problem in OSPF Networks*, GLOBECOM 2001, San Antonio, USA, 25–29 November, 2001, pp. 1605–1609

- [C12] G. Salamon, S. Györi, J. Harmatos, T. Cinkler: *Dimensioning WDM-based Multi-layer Transport Networks with Grooming by Genetic Algorithm*, 7th European Conference on Networks & Optical Communications, NOC 2002, Darmstadt, Germany, 18-20 June, 2002
- [C13] J. Harmatos: *Planning of UMTS Core Networks*, 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2002, Lisboa, Portugal, September 15-18, 2002