



M Ű E G Y E T E M 1 7 8 2

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

Anda Géza Pál

**NEMZETI OPTIKAI
GERINCHÁLÓZAT
OPTIMALIZÁLÁSA**

TDK dolgozat

KONZULENS

Dr. Cinkler Tibor

BUDAPEST, 2015

Tartalomjegyzék

1. Bevezető	3
2 Elméleti áttekintő.....	4
2.1 Többszörös hozzáférésű közegek	4
2.2 Optikai átviteli közeg.....	4
2.3 DWDM	6
2.4 Hálózatok felépítése.....	8
2.5 Védelmek	10
2.5.1 Egyéni védelem.....	11
2.5.2 Megosztott védelem.....	11
2.6 Fogalmak	11
2.7 Teszt hálózat	12
3 Kezdeti állapot.....	14
3.1 Hálózat szerkezete	14
3.2 Adathalmaz bővítése.....	15
3.3 Előzetes eredmények	16
3.4 Konkrét feladat	16
4 Kidolgozott algoritmusok és eredményeik	18
4.1 Hullámhosszok szerinti átrendezés	18
4.1.1 Felülről töltő mohó módszer.....	18
4.1.2 Alulról töltő mohó módszer	20
4.1.3 Heurisztikus módszer.....	21
4.2 Eredmények	22
4.3 <i>Védelmek</i>	Hiba! A könyvjelző nem létezik.
4.4 Eredmények dokumentálása	25
5 Összefoglalás, jövőbeni lehetőségek	26
5.1 Specifikáció teljesülése.....	26
5.2 Tapasztalatok összefoglalása	26
5.3 Jövőbeni lehetőségek	27
Irodalomjegyzék.....	28

1. Bevezető

A társadalom információfogyasztása napról napra növekszik. Optikai hálózatok régóta felelnek azért, hogy az információ nagy mennyiségben és gyorsan tegyen meg nagy távolságokat. Ám ezeknek a hálózatoknak is véges a kapacitása, új kábelek lefektetése viszont költséges és időigényes. A kutatók régóta arra törekcsenek, hogy a kihasználtságot maximalizálják, az átviteltechnikai korlátokat pedig kitolják újabb modulációkkal, kódolási eljárásokkal.

Munkám során egy nemzeti optikai gerinchálózat optimalizálásával foglalkoztam. Jelenleg az optikai átviteli közegben előre definiált hullámhosszokon történik az átvitel. Olyan különböző módszerek hatékonyságát vizsgáltam, mint a hullámhosszmódosítás mohó és adaptív algoritmusokkal, vagy elvezetési útvonalak módosítása. Az útvonalak módosítása során törekedtem arra, hogy az új elvezetések egymástól függetlenek legyenek. Ennek célja az volt, hogy egy esetleges kiesés az üzemi működést ne zavarja meg, azaz a hálózat redundáns legyen. Ezen módszerek segítségével egyszerűen és hatékonyan lehet növelni egy hasonló, kiterjedt hálózat kihasználtságát.

2 Elméleti áttekintő

2.1 Többszörös hozzáférésű közegek

Többszörös hozzáférésű (Divided Multiple Access) közegekben az átvitel felosztása jelen ismereteink szerint három módon történhet:

1. TDMA: Időalapú osztás, azaz az átviteli közeg időablakokra van osztva, amin a felhasználók osztoznak. Egy időablak egy felhasználóhoz tartozik.
2. CDMA: Kódosztás, azaz az átvitelre szánt információt speciális módon kódolják. Így a felhasználók azonos fizikai paraméterekkel osztozhatnak ugyanazon a közegen. Az ilyen hálózatokban lévő végeseleközök a több, kevert jelből dekódolni tudják az átvitt információt.
3. WDMA: Frekvenciaosztás, azaz az átviteli közeg diszkrét frekvenciatartományokra van osztva. Egy frekvenciatartományhoz egy felhasználó tartozik.

A továbbiakban mivel nem a felhasználó, hanem a hálózat oldaláról közelítem meg a témát, így többszörös hozzáférés helyett annak megvalósítási (multiplexálási) stratégiájáról fogok írni. (Divided Multiple Access -> Division Multiplexing)

2.2 Optikai átviteli közeg

Az információéhség és a felhasználói kultúra változását a technológiának is követnie kellett. Megjelentek interaktív tartalmak, így a műhold alapú adattovábbítás már nem felelt meg a célnak. Az adatmennyiség növekedésével (szöveg, hang, majd videó) olyan átviteli közegre volt szükség ami nagy távolságban, gyorsan képes nagy mennyiségű adatot továbbítani.

Az optikai közegen történő adattovábbítás már a XX. század derekán felmerült, de a csillapítás használható mértékűre csökkentése csak az elmúlt néhány évtizedben történt meg. Ahogy annak idején a digitális adattovábbítás a gerinchálózatokban jelent meg

először, úgy az optika használata is onnan indult. Az igények viszont további terjedést várnak el, így ma már nem csak a gerinchálózatokban, hanem városrészekben (FTTN – Fiber to the neighbour), házakban (FTTB – Fiber to the building) vagy akár otthonunkban (FTTH – Fiber to the home) is elérhető az optikai hozzáférés.

Ennek okai [3]:

- Hatalmas adatátviteli sebesség (jelenlegi sebességrekord 5,1 Tbps/hullámhossz [10])
- Nagy sávszélesség (közel 60 THz, viszonyítási alapként a koaxiális kábel 12.4 GHz [12])
- Alacsony csillapítás (0.2dB/km, viszonyításként 13 km hosszú optikai kábel csillapítása megegyezik 1 m koaxiális kábel csillapításával [12])
- Magas ár/kapacitás arány

Optikai átviteli technológiákkal az ITU-T külön csoportja foglalkozik [6].

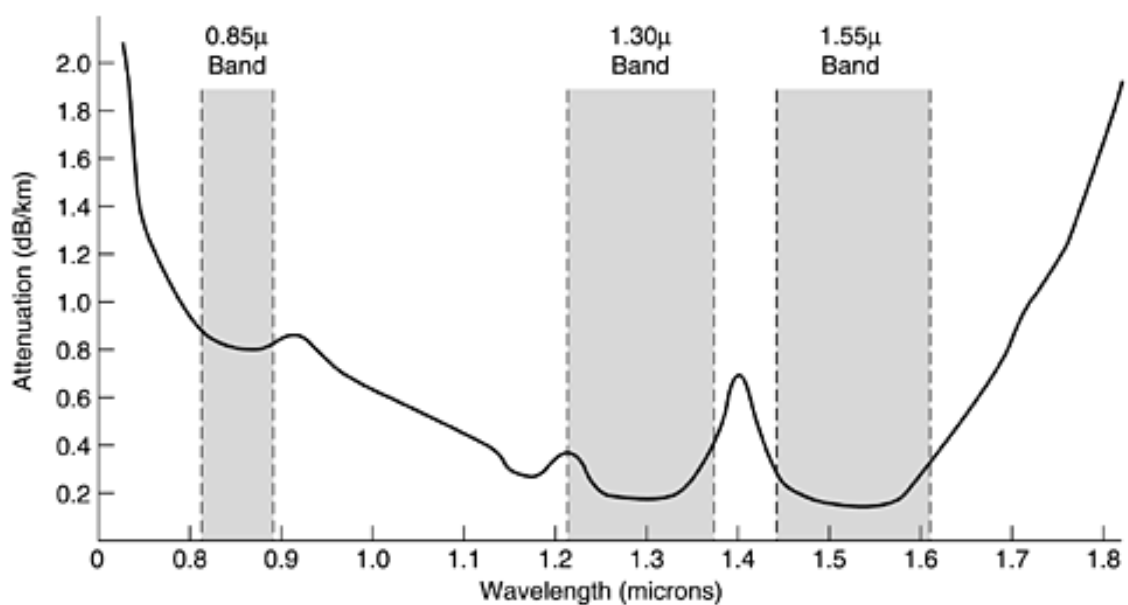
Fizikailag az átvitelre legalkalmasabb az egymódosú szál. Lényegesen kisebb magátmérője van, mint a multimódosú kábelnek, ellenben sokkal jobban tűri a csillapítást. Ezek a szálak a közönséges műanyag borítás helyett Nátrium-szilikát (vízüveg) borítással rendelkeznek. A fény által használt belső rész vastagsága 9 μm , szemben a multimódosú kábellel, amiben 50 μm .

Optikai közegen az időalapú multiplexálás nehezen megvalósítható. Ennek feltétele, hogy az összeköttetés végén lévő két feldolgozó eszköz azonos órajellel működjön és tökéletesen legyen szinkronizálva. Belegondolni is nehéz, hiszen egy hullámhossz alig több, mint 5 fs ($5,178 \cdot 10^{-15}$ másodperc), ami nagyságrendileg egy szinten van a céziumos atomóra pontosságával (10^{-16}). Arról nem is szólva, hogy ha ötvözzük hullámhosszosztással, akkor a különböző hullámhosszokon különböző időben érnek be a jelek, ami szintén problémát okoz. A kódosztásos multiplexálás szintén nehezen megvalósítható, hiszen a kódolás feldolgozásához használatos mai processzorok lényegesen lassabban működnek, mint egy ilyen átviteli közegben a jel chip-rátája [3].

Ezek indokolják azt, hogy ma az optikai átviteli technológiák alapvetően hullámhosszosztásra építenek.

2.3 DWDM

A DWDM (dense wavelength-division multiplexing – sűrű hullámhossz-osztásos multiplexálás) technológia alapjaként az ITU telekommunikációs szabványtestület mérések alapján három hullámhossztartományt határoz meg ajánlásként használatra (1. ábra: Csillapítás optikai átviteli közegben hullámhossz függvényében).



1. ábra: Csillapítás optikai átviteli közegben hullámhossz függvényében

Legjobb átviteli jellemzőkkel az 1260-1675nm-es tartomány rendelkezik. Ezt további hat különböző tartományra osztották fel [5], melyen belül a C sáv 1530–1565nm között található (ebben a sávban gyakorlatilag 0 a szálon belüli diszperzió [8]). A C (common vagy conventional) sávon belül a 193.1 THz-hez kötve frekvenciatáblázatot adott ki [7] (1552.5244nm középponti hullámhossz). A táblázatban nominális középponti hullámhosszok vannak megadva, illetve a hozzájuk tartozó nominális átviteli frekvenciák. E mellé 4 sáv szélességet jelölnek meg: 12.5, 25, 5, és 100 GHz (vagy a fölötti). Erre példa a 2. ábrán alább látható. Az, hogy milyen sáv szélességet használ egy-egy szolgáltató, határozza meg, hogy az ajánlott tartományt hány részre osztják fel.

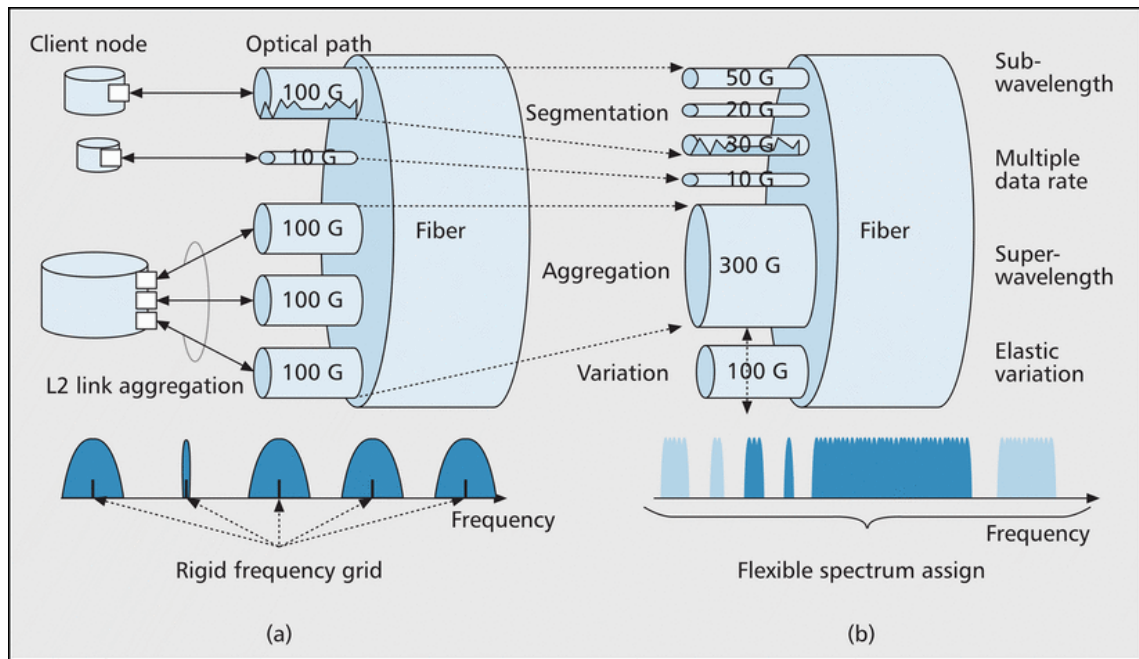
Nominal central frequencies (THz) for spacings of:				Approximate nominal central wavelengths (nm) (Note)
12.5 GHz	25 GHz	50 GHz	100 GHz and above	
193.1000	193.100	193.10	193.1	1552.5244
193.0875	–	–	–	1552.6249
193.0750	193.075	–	–	1552.7254
193.0625	–	–	–	1552.8259
193.0500	193.050	193.05	–	1552.9265
193.0375	–	–	–	1553.0270
193.0250	193.025	–	–	1553.1276
193.0125	–	–	–	1553.2282
193.0000	193.000	193.00	193.0	1553.3288
•	•	•	•	•
•	•	•	•	•

2. ábra: ITU-T G694.1 ajánlás nominális hullámhosszokra és frekvenciákra -részlet -

Ezen ajánlás alapján a C sávot 100 GHz sávszélességgel fel lehet osztani 72 különböző hullámhosszcsatornára. Ez a felosztás „atomi” része, egy ilyen hullámhosszcsatornához egy átvitel tartozik.

Ez az úgynevezett ITU „grid” fix távolságokban határozza meg a nominális középponti hullámhosszokat. Az egy hullámhosszra ültetett vonal sávszélességét a vonal két végpontján található eszközben lehet konfigurálni. Természetesen két különböző vonal között frekvenciában védősávot kell hagyni, ezért ritkul a nominális hullámhosszok száma a frekvencia növelésének függvényében.

Ezzel kapcsolatban érdemes megjegyezni, hogy nem minden hálózat fix hullámhosszkiosztásra épül. Kutatások során felfedezték, hogy a rugalmas hullámhosszosztás (EON – Elastic Optical Network) [14] ugyan komplexebbé teszi a hálózatot (és drágábbá a kiépítés költségét), ugyanakkor növekszik a kihasználtság, ezáltal a kapacitás. Az alább látható ábra ezt próbálja szemléltetni. Baloldalon látható, hogy az egyes vonalak középpontja (nominális középhullámhossz) egyenlő távol esik a többitől. A jobb oldalon látható flexibilis kiosztás viszont egyedi megoldásokat tesz lehetővé, ami által a hálózatot jobban ki lehet használni. A munkám során fix kiosztású hálózaton dolgoztam, így a továbbiakban erről lesz szó.

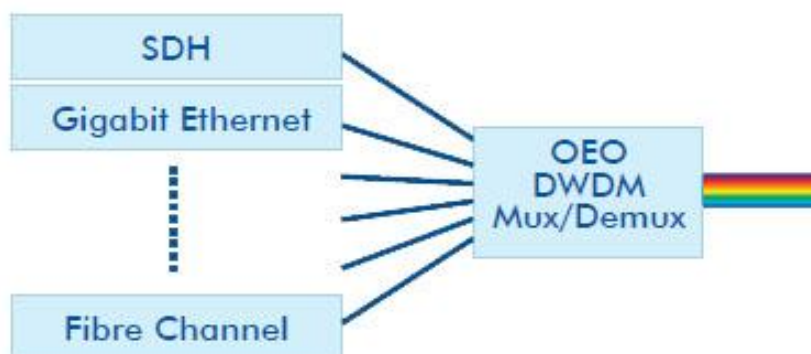


3. ábra: Fix és rugalmas frekvencia kiosztás

Az optimalizálásra és a kihasználtság növelésére való törekvés aktualitását jól mutatja, hogy a probléma számos országban fellelhető. Az AIT kutatóintézet évek óta foglalkozik a Francia gerinchálózat bővítésének lehetőségeivel. [15] Az igények növekedésével a hálózat néhány éven belül el fogja érni a technológia fizikai korlátait. Kutatásaik alapján lehetőségek még az előbb említett EON típusú hálózatokban, illetve a spektrális rugalmasságban vannak.

2.4 Hálózatok felépítése

Optikai hálózatokat jellemzően olyan helyeken használnak ahol két egymástól távol álló hálózatot kell összekötni vagy két hálózat között nagy forgalmat kell lebonyolítani. Mindkét esetben szükséges, hogy a lokális, elektronikus átvitelen alapuló hálózatból optikai hálózatba lépéskor átalakítás történjen. Ezt az átalakítást úgynevezett E/O illetve O/E (eletrical to optical illetve optical to eletrical) jelátalakítók végzik. Az előbbi a bejövő elektromos jelet optikai jellé alakítja, míg az utóbbi az optikai jelet alakítja elektromos jellé (4. ábra). A gyakorlatban a folyamatot végeztetésnek, a folyamatot végző eszközt pedig transzponder hívják. A transzponder komplexitását az adja meg, hogy hány különböző hullámhossz érzékelésére és kezelésére alkalmas.



4. ábra: Elektronikus és optikai hálózat kapcsolata (transzponder) [5]

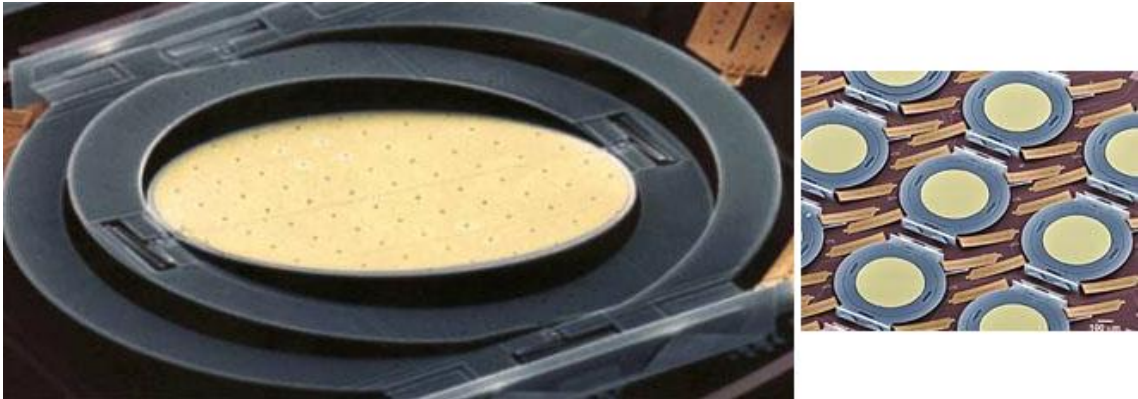
Az átalakítás mellett szükség van az adatok hullámhosszcsatornákra történő rendezése. A már fényként közlekedő (E/O által) adatok egy optikai multiplexer által kerülnek egy fényszálra. A szál másik végén a folyamatot ellentétes irányban egy optikai demultiplexer végzi, szétválasztja a hullámhosszokat. A külön álló jeleket adott hullámhosszokra szintezett O/E transzponder alakítja vissza elektromos jellé, ahogy az az 5. ábraán is látható.



5. ábra: Optikai multiplexer és demultiplexer [5]

Ha a bekezdés elején megfogalmazott gondolatot megfordítjuk és az elvet nem pont-pont összeköttetésekre, hanem gerinchálózatokra alkalmazzuk, akkor újabb feladatok merülnek fel. Ha egy hálózati csomópont több optikai kimenettel is rendelkezik, akkor szükséges lehet a fénynyalábok továbbítása egyik ágból a másikba (kapcsolás - transzfer). Előfordulnak olyan esetek, hogy a bejövő és a kimenő ágon eltérnek a hullámhosszok (hullámhossz-konverzió). Komplexebb csomópont esetén (2-nél több ágat kezel) szükséges lehet az irányok közül választani és ezek között kapcsolni (útvonalválasztás).

A kapcsolás és útválasztás feladatát szintén optikai eszközök látják el, hiszen az elektronikus elven működő eszközök nem tudnák ellátni ezt a feladatot. Fizikailag a kapcsolást apró tükrökkel oldják meg. A tükör a névleges hullámhosszból kiindulva igen kicsi és még mozognia is kell. Ezeket az eszközöket mikroelektronikus-mechanikus rendszernek hívják (MEMS).



6. ábra: MEMS tükör és tükrökből álló kapcsolótábla [13]

Egy-egy ilyen tükör átmérője akár 0,8 mm is lehet, irányzópontossága pedig eléri a $0.5\mu\text{m}$ -t (viszonyítás képpen egy emberi hajszál $0,05\text{mm}$). Ezekből a tükrökből készítenek lapokat (amik közel $0,5\text{cm}$ alapterületűek, vastagságuk kevesebb, mint $1\mu\text{m}$) [13]. A tényleges modult, ami a kapcsolást végzi, OXC-nek (Optical Cross Connect – Optikai keresztkapcsoló) hívják. Az olcsóbb OXC modulok hullámhossz-konverzióra nem képesek, tehát az adott nyalábot csak továbbítják, hullámhosszán módosítani nem tudnak. A gyakorlatban ez azt eredményezi, hogy az adatok két végpont (végeztetés) között végig ugyanazon a hullámhosszon haladnak.

A DWDM alapú teszhálózaton a csomópontok hullámhossz konverzió nélkül továbbítják a jelet útvonalválasztás nélkül. Az útvonalak tehát előre vannak huzalozva (passzív router).

2.5 Védelmek

A kapacitás mellett a megbízhatóság is fontos szempont egy hálózat tervezése során. Ahogy az egészségügyben, vagy a közlekedésben, úgy távközlési szektorban sem megengedhető egy szolgáltatás kiesése. 100%-os rendelkezésre állás szükséges, hiszen az úgynevezett három kilences rendelkezésre állás (99,999%) is körülbelül 8 perc kiesés

jelent éves szinten. Jól jelzi az erre való felkészült, hogy egy Tier1 szolgáltató esetén a hálózat kihasználtságának 5-10%-os túllépését már túlterhelésnek minősítik. A hálózat részére a sérthetlenség nem garantálható. Építkezések és háborúk mindig vannak, amiknek vele járója, hogy egy-egy összeköttetés megszakadhat. A védelmek célja, hogy az ezzel járó kiesést pótolja. Védelmi útvonalak alapvető feltételi, hogy a hozzá tartozó üzemi úttól teljesen független legyen.

2.5.1 Dedikált védelem

Dedikált védelem esetében minden üzemi út mellé rendelünk egy védelmi utat. Ez azt jelenti, hogy az üzemi út mellé fenntartunk egy olyan link független alternatív útvonalat, amit csak az üzemi út használhat és csak kiesés esetén. Mivel ezek a védelmi utak előre konfigurálva vannak, az átállás másodpercek alatt meg tud történni, akár automatizáltan is. [17]

2.5.2 Megosztott védelem

A védelmek másik fajtája az úgynevezett megosztott védelem. Ez azt jelenti, hogy egy definiált védelmi útvonalon több üzemi útvonal is osztozik. Amennyiben valamelyik üzemi út kiesik, akkor az üzemi csoport részére fenntartott védelmi útvonal pótolja a kieső kapacitást. Egy későbbi alkalommal, mikor egy másik üzemi út esik ki, azt ugyanúgy tudja helyettesíteni. Ez a megoldás komplexebb a dedikált védelemnél, ugyanakkor sokkal kisebb erőforrás lefoglalását igényli. Fontos megjegyezni, hogy ez a példa egy összeköttetés hibája esetén képes pótolni a kiesést, két hiba esetén a megosztott védelem ebben a formában kevés. [17]

2.6 Fogalmak

Egy hálózat működtetése során felmerülő költségeket OPEX-nek (operational expense), míg a fejlesztésekhez kapcsolódó költségeket CAPEX-nek (capital expenditure) hívja a szaknyelv. Működő hálózatok esetén értelemszerű cél, hogy a működéshez szükséges költségeket minimalizáljuk, ehhez pedig minél kevesebb fejlesztési költséget hozunk létre.

Egy konkrét optikai hálózat általánosan modellezhető gráfokkal, így a következő fogalmakat vezetem be, melyek a továbbiakban a megértéshez szükségesek:

- csomópont (node): A gyakorlatban egy olyan központ, ahol transzponderek és oxc-k vannak.
- él (link): Két csomópontot összekötő optikai kábel.
- port (iface): Az optikai kábelek konkrét végpontja egy csomópontban. Egy csomópontnak több (legalább kettő) portja is van. Minden porthoz tartozik egy kapcsoló, ami az adott hullámhossz jelét egy másik portra kapcsolja (transzfer igény) vagy végezteti a transzponderen.
- igény (demand): két csomópont között teljes hullámhosszcsatornát lefoglaló elvezetés
- útvonal (route): éllista, ami megadja, hogy az igény a két végeztető csomópont között melyik linkeken halad keresztül.
- hullámhosszcsatorna (röviden hullámhossz, vagy csatorna): az a hullámhossz, amin az igény elvezetésre kerül.
- hullámhossz gráf: egyértelműen azonosítja az igény elvezetését, azaz megadja a relációt útvonal és hullámhossz között [1].

2.7 Teszt hálózat

Kutatásom során egy nagy magyar telekommunikációs vállalat optikai gerinchálózatát vettem alapul. Tekintve hogy a hálózat a vállalat ipari titkát képezi, így én is csak a kutatáshoz szükséges legfontosabb részletekhez jutottam hozzá.

Az említett hálózat létrehozásakor 40 hullámhosszcsatorna került meghatározásra. A legutóbbi kapacitás növelés során a hálózati csomópontok egy részére olyan eszközök kerültek, melyek kétszer annyi hullámhosszcsatornát képesek kezelni. A 2.3 DWDM részben tárgyalt okokból az ITU-T ajánlása alapján az új hullámhosszok a meglévők

közé kerültek. Ezáltal létrejött, 80 hullámhosszt kezelő hálózatban a kihasználtság meglehetősen egyenlőtlené vált. A továbbiakban a hullámhosszokat 0-tól 79-ig fogom számozni, az eredeti 40 hullámhosszra, mint párosakra fogok hivatkozni, míg a további 40-re, mint páratlan hullámhosszok.

A kapacitásnövelés másik fázisaként aggregáltak igényeket. Ennek apropóját az szolgáltatta, hogy ugyanazon csomópontok között, ugyanazon útvonalon több 10 Gbps vonal is volt definiálva, amelyek egyenként lefoglaltak egy-egy hullámhosszcsatornát. Ezen adatforgalom egy darab 100 Gbps-os vonalra aggregálásával számos hullámhosszcsatorna felszabadítható (a nominális középfrekvencia változatlan marad, a sáv szélességet viszont a két végpontban megváltoztatják, ezáltal növekszik az átvihető adat mennyisége). Ezen különböző sáv szélességű vonalak szomszédos hullámhosszon való vezetése esetén az áthallás miatt komoly adatvesztést szenvedtek. Mérések alapján megállapították, hogy egy 10 Gbps-os és egy 100 Gbps-os vonal között legalább 6 hullámhossznyi szabad sávot kell hagyni. A további átalakításoknak tehát ezt figyelembe kell vennie.

Az előbb említett problémára kézenfekvő megoldás, hogy a teljes hullámhossztartományt két részre osztjuk fel, melyet egy 6 csatorna széles sáv választ el egymástól. A tartomány egyik felére a jelenlegi 10 Gbps-os vonalak kerülnek, míg a másik tartományba lehetőség van jövőbeni 100 Gbps-os igények definiálására.

Kutatásom során erre a problémára próbáltam megoldásokat keresni.

3 Kezdeti állapot

3.1 Hálózat szerkezete

A hálózatot leíró adathalmaz négy, vesszővel tagolt fájlban álltak rendelkezésre. Ezek az alábbiak:

```
nodes (node_id)
```

Csomópontokat leíró fájl, csak az azonosítókat tartalmazza.

```
ifaces (node_id, port_id, xconn, oddwl)
```

Egy adott csomópontba befutó optikai kábel csatlakozási pontja. Paraméterei hogy melyik csomópontban található, saját sorszáma, képes-e kapcsolásra (van-e benne oxc), valamint képes-e páratlan hullámhosszok végeztetésére (transzponder képessége). Fontos megjegyezni, hogy a transzponder és az oxc egymástól független eszközök. Előfordulhat olyan csomópont ami képes ugyan a páratlan hullámhosszok kapcsolására, de végeztetni nem tudja azt. Mivel a feladat megoldása során az előre definiált igényeket nem módosítottam végpontjaik tekintetében, így a továbbiakban ezekre nem is fogok hivatkozni.

```
links(link_id, snode_id, sport_id, dnode_id, dport_id, length)
```

Konkrét optikai kábeles kapcsolat két csomópont között. Paraméterei az azonosítója, az általa összekötött két csomópont (source és designation) és az ott kapcsolódó interfész azonosítója, valamint a saját hossza.

```
demands(demand_id, snode_id, dnode_id)
```

Két tetszőleges csomópont közötti adatkapcsolat. Paraméterei az azonosítója, valamint a két csomópont azonosítója, amik között működik.

```
routes(demand_id, seq, link_id, wl)
```

Adott igény konkrét útvonala. Paraméterei az igény azonosítója, az útvonalon belül a kiinduló csomóponttól hányadik él, magának az érintett élnek az azonosítója, valamint hogy az igény melyik hullámhosszon halad.

A rendelkezésre álló adatok alapján 48 csomópont van a hálózatban, közöttük összesen 67 link fut, melyeken 121 igény van elvezetve. A 121 igényből 30 olyan, amit lehetőségünk van páratlan hullámhosszon vezetni (azaz mindkét végpontjában lévő interfész képes páratlan hullámhosszt kezelni).

3.2 Adathalmaz bővítése

A fentebbi adatokból racionalizált sémákat hoztam létre, hogy azok az optimalizálás során könnyen felhasználhatóak legyenek. Ezt többek között az indokolja, hogy a jelenlegi struktúra egy bővebb adatbázis részhalmaza, melyből attribútumok törlésre kerültek. Emiatt a sémák normalizálási foka csökkent. Emellett szükséges további információkkal kiegészíteni a struktúrát a számítások megkönnyítése érdekében.

A csomópontokat további információval szükségtelen volt ellátni. Az interfész viszont egyértelmű kapcsolatban van egy linkkel, így az adathalmazba bekerült ez a paraméter.

```
ifaces (node_id, port_id, oddwl, link_id)
```

A linkeket leíró relációt is több információval bővítettem. Feljegyzésre került, hogy hány foglalt hullámhossz van a linken, valamint minden elemhez hozzáfűztem egy újabb relációt. Ez a reláció valójában egy lista volt a linken vezetett igényekről a hullámhosszuk függvényében.

```
links(link_id, snode_id, sport_id, dnode_id, dport_id, length,  
wl[], num_of_occupied_wl)
```

A legtöbb információt az igényekhez számoltam hozzá. Hozzáfűzésre került az igény hullámhossza, hogy a két végpontja képes-e páratlan hullámhosszokat kezelni, hány különböző szabad hullámhossz van az útvonal (amire áthelyezhető), a hossza, lépésszáma, valamint 3 újabb reláció. Az egyik azon linkek listája, amin áthalad az igény, a második azon igények listája, melyek útvonalában függetlenek az adott

igénytől (nem haladnak át azonos linken), végül a harmadik relációba kerültek azok az igények, amelyek más útvonalon haladnak, de azonos a két végpontjuk.

```
demands(demand_id, snode_id, dnode_id, wl, could_odd_wl,  
wl_chance, lenght, hops, link_list[], dependent_demands[],  
same_demands[])
```

Az útvonalakat leíró reláción nem módosítottam.

3.3 Előzetes eredmények

Dr. Szigeti János hasonló alapokról kereste a hálózatban az optimumot. Ő különböző engedményekkel (befektetett erőforrásokat nem számolva, módosítás lépéseit nem követve) kereste a hálózat végső kapacitását. Kidolgozott olyan algoritmusokat, melyek többek között képesek voltak a rendelkezésre álló igényeket a lehető legkevesebb hullámhosszon elvezetni vagy éppen egyenletesen egy megadott tartományban. Eredményei alapján arra jutott, hogy a számomra is rendelkezésre álló információk alapján az igények mindössze 23 hullámhosszcsatornán elvezethetőek [2]

Természetesen ezek az információk a gyakorlatban dimenzionálás céljából hasznosak, de a konkrét eredmények valóságba átültetése ebben a formában megvalósíthatatlan (tulajdonképpen a hálózat teljes leállításával járna, ami pedig ütközik az alapfeltétellel).

3.4 Konkrét feladat

A feladat megoldása szempontjából felmerült további konkrétumok:

- A változásokat minél kevesebb lépésben kell alkalmazni. Egy igény hullámhosszának a módosítása egységnyi munkával jár, így törekedni kell arra, hogy egy igény áthelyezésekor az már rögtön a megfelelő helyre kerüljön. Ez azt is jelenti, hogy az optimalizáló algoritmus bonyolultsága tetszőleges lehet, az egyetlen optimalizálási paraméter a módosítások száma.

- Működésben ne okozzon fennakadást a hálózat átalakítása. Ez azt jelenti, hogy egy hullámhosszt nem szabadíthatunk fel, amíg a rajta lévő igény nem került teljesen áthelyezésre (smooth transition).
- A meglévő csomópontoknak csak egy része képes 80 hullámhosszcsatorna végeztetésére, így figyelni kell arra, hogy bizonyos csomópontok csak minden második (páros) hullámhossz kezelésére alkalmasak.
- A rendelkezésre álló nyolcvan hullámhosszcsatorna három részre legyen osztva. Egyik felére a 100Gbps sávszélességű igények fognak kerülni a jövőben, másik felére az aktuális 10Gbps sávszélességű igények, míg köztük 6 hullámhosszcsatornányi védősávot kell hagyni. Hogy a határ hova kerüljön, az az optimalizálás függvényében dől el.

Ezen kritériumok mellett a hullámhosszokat 0-tól 79-ig számoztam, a határvonalat pedig változóként kezeltem és ennek a minimumát kerestem, az igényeket pedig a határpont fölé próbáltam elhelyezni.

4 Kidolgozott algoritmusok és eredményeik

4.1 Hullámhosszok szerinti átrendezés

Elgondolásában három különböző módszert készítettem a feladat végrehajtására. Mindegyik a megadott adatstruktúrán dolgozik, bemenő paramétere a tartomány határa, kimenete az új elrendezés. A különböző módszereken belül is több algoritmust készítettem, melyek különböző dolgokat vesznek figyelembe.

Első lépésként kézenfekvőnek tűnt, hogy az optimalizálásra valamilyen mohó algoritmus adaptációt alkalmazzak. Végül két megoldás mellett döntöttem. Az egyik a kijelölt tartomány minimumától kezdi el inkrementálisan a feltöltést, míg a másik a maximumától dekrementálisan. Ezen felül további attribútumok számítása után egy ezekből heurisztikusan optimalizáló algoritmust is írtam.

Az eredményeket két további paraméterrel próbáltam megkülönböztetni. Az egyik ilyen paraméter a lépésszám. Minden egyes igény hullámhosszának megváltoztatása egy lépésnek minősül. A másik mérőszám a futásidő. Egy változóban feljegyzem az időpontot, amikor a program elindul, tolsó lépésként pedig összehasonlítom az akkori időponttal. A kettő különbsége fogja adni a teljes algoritmus futásának idejét.

4.1.1 Felülről töltő mohó módszer

Az algoritmus nem kapja meg paraméterként a határt, hanem először feltételezi, hogy a határ a 79-es hullámhossz. A határt iterációnként csökkenti és próbálja meg a határ alatt lévő igényeket a határ fölé helyezni. Amikor már nincs a határ alatt igény, akkor az algoritmus megáll.

A verzió:

```
border:=79

while(below_border)
  for(∀demand|demand->wl<border)
    if(demand->is_odd_wl)
      for(wl:=79 to border step by -1)
```

```

        route_is_free = true;
        for(demand->route)
            if(!is_free(route->link))
                route_is_free:=false
    else
        for(wl:=78 to border step by -2)
            route_is_free = true;
            for(demand->route)
                if(!is_free(route->link))
                    route_is_free:=false
    if(route_is_free)
        demand->wl:=wl
        break
border := border - 1

```

Minden igényt, ami a határon kívül halad, sorra veszünk. A második ciklus a jó hullámhosszokon halad végig. Ha képes az igény páratlan hullámhosszon végződni, akkor 79-ről indul és csökken egyesével, ha nem akkor 78-ról csökken kettesével. Minden hullámhossz esetén megvizsgáljuk az igény teljes útvonalát és hogy ott már használatban van-e az adott hullámhossz. Ha szabad a hullámhossz, akkor áthelyezzük oda az aktuális igényt.

B verzió:

```

border:=79

while(below_border)
    for(∀demand|demand->wl<border)
        if(demand->is_odd_wl)
            for(wl:=79 to border step by -2)
                route_is_free = true;
                for(demand->route)
                    if(!is_free(route->link))
                        route_is_free:=false
        else
            for(wl:=78 to border step by -2)
                route_is_free = true;
                for(demand->route)
                    if(!is_free(route->link))
                        route_is_free:=false
    if(route_is_free)
        demand->wl:=wl
        break
border := border - 1

```

A második algoritmus annyiban tér el az elsőtől, hogy a páratlan hullámhosszokon végeztethető igényeket csak páratlan hullámhosszra engedi áthelyezni. Ezáltal több esélyt teremtve a csak páros hullámhosszon végeztethető igényeknek.

C verzió:

```
border:=79
p = 1;
while(below_border)
  for(∀demand|demand->wl<border)
    if(demand->is_odd_wl & p = 1)
      for(wl:=79 to border step by -2)
        route_is_free = true;
        for(demand->route)
          if(!is_free(route->link))
            route_is_free:=false
        if(route_is_free)
          demand->wl:=wl
          break
      border := border - 1

p := 2;
border := 79
while(below_border)
  for(∀demand|demand->wl<border)
    if(!demand->is_odd_wl & p = 2)
      for(wl:=78 to border step by -2)
        route_is_free = true;
        for(demand->route)
          if(!is_free(route->link))
            route_is_free:=false
        if(route_is_free)
          demand->wl:=wl
          break
      border := border - 1
```

A harmadik esetben először csak azokon a hullámhosszokon halad végig az algoritmus, amelyek áthelyezhetőek páratlan hullámhosszra és csak oda teszi át őket. Ezután veszi sorra a többi igényt és helyezi el a már előbb látott módon.

4.1.2 Alulról töltő mohó módszer

Ezen módszer esetében az algoritmus paraméterként megkapja a határt. Végigjárja az igényeket és alulról felfelé haladva az első alkalmas hullámhosszra helyezi át az igényt.

```
route_is_free:=true
border:=43

for(demand|demand->wl<border)
  for(wl:=80 to border step by -2)
    for(demand->route)
      if(!is_free(route->link))
        route_is_free:=false
    if(route_is_free)
```

```
demand->wl:=wl  
break
```

Különbség az eredmények terén mégis várható, hiszen a hálózat eredeti állapota véletlenszerűnek mondható.

4.1.3 Heurisztikus módszer

Ahhoz, hogy a módosítást ésszerűbben el tudjuk végezni, további mellékszámításokra van szükség. Az adatstruktúrában már számon van tartva, hogy egy igény aktuálisan hány különböző hullámhosszra kerülhet át az útvonal módosítása nélkül. Emellett szükség van arra is, hogy az egyes linkeken belül a még szabad hullámhosszokra hány különböző igény kerülhet át. Erre azért van szükség, mert elképzelhető, hogy lesznek telített és kevésbé telített hullámhosszok.

A módszer lényege, hogy minden iterációban megkeressük azt az igényt, amelyiket a legkevesebb különböző hullámhosszra lehet áthelyezni. A kiválasztott igény számára elérhető hullámhosszok közül hozzá vesszük azt, amelyikre a legkevesebb igény kerülhet át és oda át is helyezzük.

Egy igény áthelyezésével más igények elől vesszük el a lehetőségeket. Az iterációk előrehaladtával a *wl_chance* paraméter értéke egyre kisebb lesz. Ezért választjuk mindig a legkevesebb eséllyel rendelkező igényt az áthelyezésre.

Ennek a számításnak két célja is van. Az egyik hogy az áthelyezéseket minél kevesebb lépésben oldjuk meg. Előfordulhat olyan eset, mikor egy igényt feleslegesen részesítünk előnyben és úgy helyezzük át, hogy egy másik igény lehetőségeit szűkítjük. A másik cél hogy az áthelyezés után a hullámhosszak szerinti elosztásban egyenletesebb képet kapjunk, mint a mohó algoritmusoknál, ahol várhatóan elég egyenetlen lesz az elosztás.

Ha az optimalizálandó tartomány túl kicsi lenne, előfordulhat, hogy olyan igényekhez jutunk, amikhez tartozó *wl_chance* érték 0. Ez azt jelenti, hogy nincs olyan hullámhossz, ahova át lehetne helyezni az igényt. (ezeket az igényeket a továbbiakban lehetetlen igénynek fogom hívni).

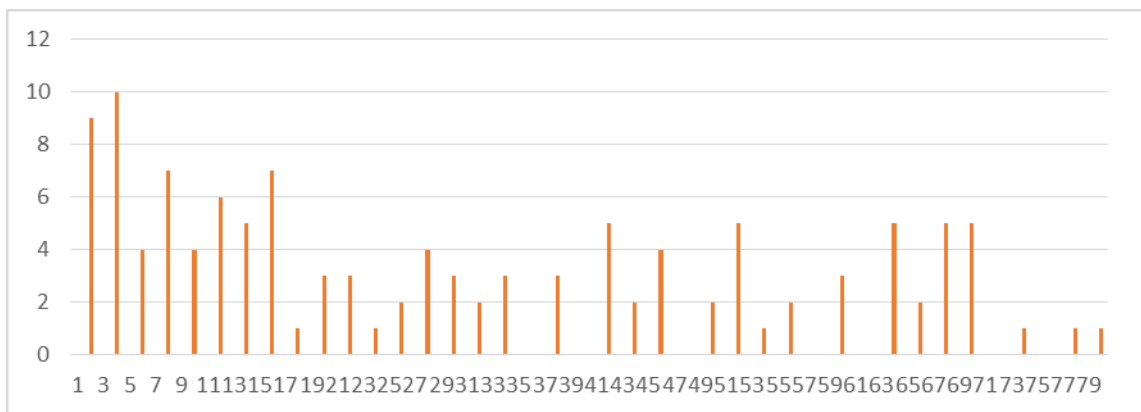
Jelenleg már minden részlet rendelkezésre áll, így a teljes algoritmus így néz ki:

```
calculate_chances()
while(have_demands_below_border)
  demand_to_change := min(demands->wl_chance)->demand_id
  wl_to_change := min(count(free_wl_list[]))
  change_demand_to_wl(demand_to_change, wl_to_change)
  recalculate_chances()
```

Ez az algoritmus jelen formájában nem foglalkozik a lehetetlen igényekkel.

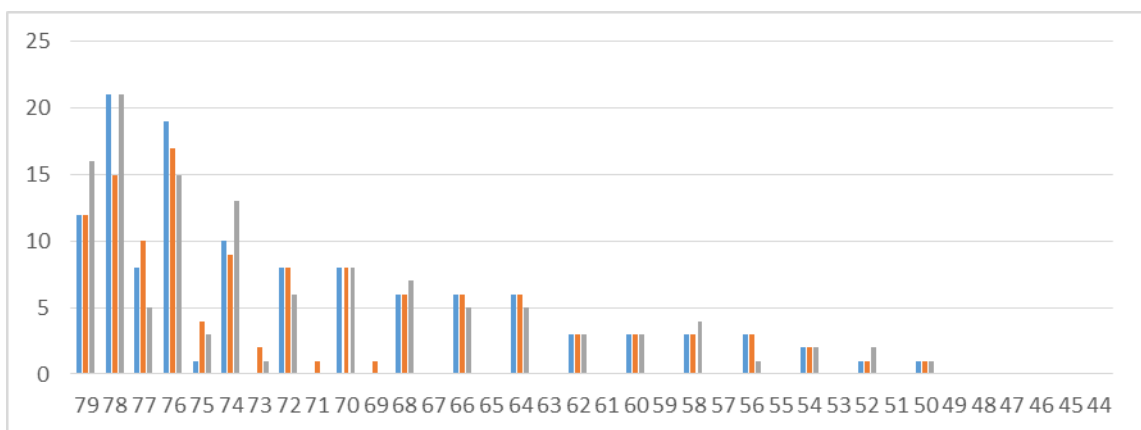
4.2 Eredmények

Kiinduló helyzetet az alábbi ábra mutatja:



7. ábra: Hálózat igényei a különböző hullámhosszokon

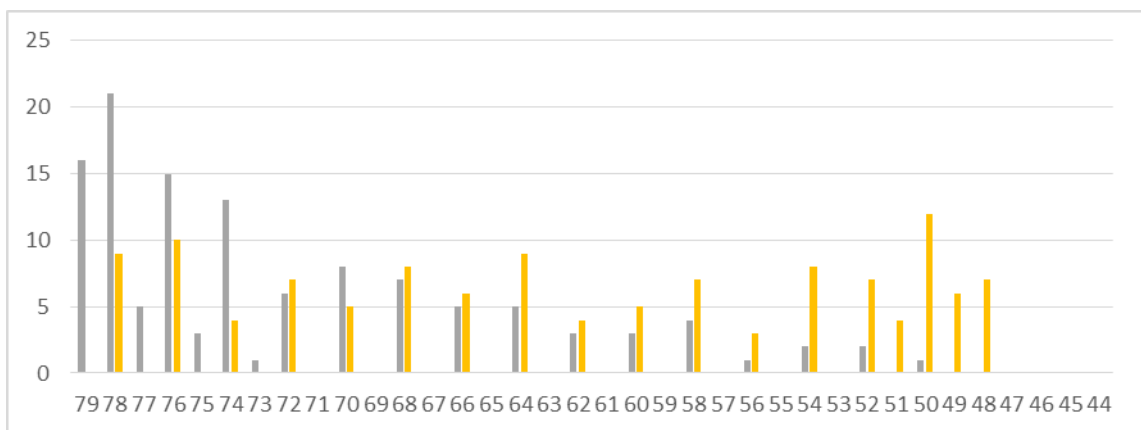
Az felülről töltő algoritmusok különböző verzióinak eredményét a 8. ábra szemlélteti.



8. ábra: Felülről töltő algoritmusok eredményei (A-kék, B-piros, C-zöld)

Látható némi eltérés a különböző algoritmusok között. Mindhárom esetben jól látható, hogy az algoritmus először a magasabb hullámhosszokon próbált helyet keresni az igénynek. Ezért látható az eredményekben az eltolódás.

Hasonló jelenség figyelhető meg az alulról töltő algoritmus esetében, ami összehasonlításképp a felülről töltő módszer C algoritmusával együtt látható a 9. ábraán. Az egyenetlenség nem olyan szembetűnő, de a páratlan hullámhosszok kihasználtságán egyértelműen látható.



9. ábra: Alulról és felülről töltő módszerek különbségének szemléltetése (szürke – felülről töltő C, arany, alulról töltő)

A négy algoritmus közötti érdemi különbséget az összefoglaló 1. táblázat mutatja.

1. táblázat: Mohó algoritmusok összehasonlítása

Algoritmus	határvonal	lépésszám	futásidő
Felülről töltő A verzió	49	94	~ 1,1 s
Felülről töltő B verzió	49	98	~ 1,5 s
Felülről töltő C verzió	49	90	~ 2,8 s
Alulról töltő	48	50	~ 0,6 s

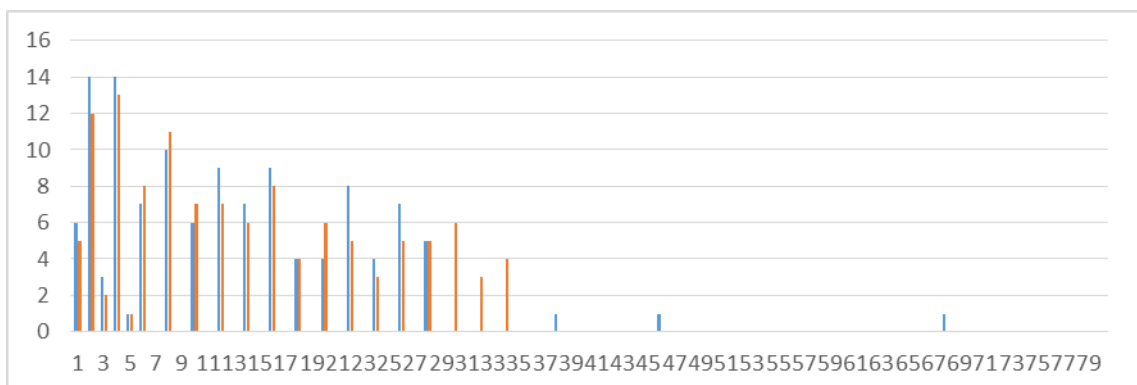
Ha először a 3 felülről töltő változatot figyeljük meg, az látható, mind ugyanazt a határvonalat találták meg különböző lépésszámmal és futásidővel.

A heurisztikus algoritmus esetében a lehetetlen igények és a határvonal kijelölése miatt több futtatást is végeztem, melynek eredményei az alábbi táblázatban láthatóak.

2. táblázat: Heurisztikus algoritmus eredményei különböző határokkal

határvonal (tényleges)	lehetetlen igények száma	lépésszám	futásidő
42 (42)	0	43	~ 22,4 s
46 (46)	0	46	~ 18,1 s
47 (48)	0	49	~ 17,8 s
48 (48)	0	49	~ 18,5 s
49 (12)	2	49	~ 18,9 s
50 (12)	2	49	~ 17,9 s
52 (12)	3	51	~ 18,4 s

Látványos különbség az 52-es és 46-os határral definiált futtatások között láthatóak az alábbi ábrán.



10. ábra: Heurisztikus algoritmus hullámhosszeloszlása 52 és 46 határral

4.3 Eredmények dokumentálása

Minden egyes futtatás során a módosításokat dokumentálni kellett a jövőbeni, lépésről lépésre történő módosítás okán. Ehhez a következő relációt alkalmaztam a hullámhosszok módosítása esetében:

```
change_log_wl(change_id, demand_id, old_wl, new_wl)
```

ahol a módosításokat azonosítóval láttam el, feljegyeztem az érintett igényt, valamint hogy melyik hullámhosszról melyikre lett áthelyezve.

A dokumentálást az eredeti formátumnak megfelelően vesszővel tagolt, fejléccel rendelkező fájlokba végeztem.

5 Összefoglalás, jövőbeni lehetőségek

5.1 Specifikáció teljesülése

A módosítások száma jól közelíti az elvi optimális korlátot már a mohó algoritmusok esetében is. Elvileg egy igény által használt hullámhossz módosítása néhány nap alatt kivitelezhető, így a teljes optimalizálás kevesebb, mint egy év alatt kivitelezhető. Ha azzal is számolunk hogy a módosítások nagy része párhuzamosítható, az eredmények igen sikeresnek mondhatóak.

A puha átállás maradéktalanul megvalósul, hiszen csak úgy módosítható egy igény hullámhosszhasználata, ha az előzőleg már szabad volt.

A többi feltételt az algoritmusok kidolgozásakor feltételként szintén beépítettem, így folyamatosan ügyeltem arra, hogy egy páratlan hullámhosszsávon közlekedő igény ne köthessen össze olyan csomópontokat, amik nem képesek azt kezelni.

Minden lépés a 4.3 Eredmények dokumentálás pont alattiak szerint részletesen feljegyzésre került.

5.2 Tapasztalatok összefoglalása

A feladat rávilágított arra, hogy az optimalizálásnak számos módja van és meglepő módon a mohó algoritmusok is lehetnek jók. Ahogy Cinkler Tibor és Szigeti János is munkájukban azt vették észre, hogy a véletlenszerűen meghatározott módosítások többször hatékonyabbak voltak a többszörösen átgondolt és irányított módosításokkal szemben.

Az algoritmusok futásidejét tekintve a két mohó módszer az igénylista szisztematikus, lineáris bejárását igényli, így lépésszáma az igények számával lineárisan becsülhető. A harmadik módszerről ez nem mondható el. Minden ciklusban kiszámolt függőségek az igények számával szintén lineárisan becsülhetőek felülről (olyannyira, hogy a futás előrehaladtával fogynak az áthelyezendő igények, a számítás gyorsasága is növekszik). Emellett a kialakult függőségek rendezése, igények ismételt bejárása és egyenkénti

áthelyezése lassítja az algoritmust. Elméletben ez még mindig felülről becsülhető az igények skalárszorásával, ám lényegesen lassabb, mint az első két változat.

Az optimalizálás során kiderült, hogy a hálózat több szempontból egyenlőtlen kialakítású, számos szempontból fejlesztésre szorul. Többek között sok útvonal kihasználatlan, aminek az oka az, hogy olyan csomópontokon halad keresztül, melyek nem tudnak kapcsolni, csak a csomópontban végeztetni az igényeket. Egy oxc beépítésével új, szinte üres útvonallal lehetne a többi terheltségét csökkenteni. A hullámhossz szerinti optimalizálásnak további korlátot ad, hogy több olyan útvonal is van, melynek végpontjai nem képesek páratlan hullámhosszok kezelésére. A csomópontok fejlesztése mindenképpen indokolt (precízebb transzponderek beépítésével), de ahogy látható, ez a hálózat biztonságának szempontjából is indokolt.

5.3 Jövőbeni lehetőségek

A kidolgozott módszerek hasznos eredményeket hoztak és a jövőben is rugalmasan felhasználhatóak a hálózat további optimalizálásakor. Ugyanakkor a számomra elérhető adatok erősen aluldetermináltak. Munkám során számos anomáliával találkoztam, ami alapvetően technológiai megoldásokkal nem volt magyarázható. Az adatok többszöri tisztítására is szükség volt. Többek között szerepelt az eredeti adathalmazban izolált csomópont (mi miatt a Dijkstra algoritmus nem működött) vagy hogy egy útvonalnak melyik is a kezdő és végpontja. Az adatok kiterjesztésével újabb paraméterek és lehetőségek merülhetnek fel az optimalizálás kérdésében.

A vizsgált hálózatban lévő eszközök egyike sem képes hullámhosszkonverzióra, azaz a csomópontban az adott igény hullámhosszát módosítani. Ez a funkció nagy mértékben növeli a hálózat komplexitását és összetettségét, ám további optimalizálási lehetőségek rejlenek benne. A fejlesztés tulajdonképpen a kapcsolatban történne, a csomópontok ettől függetlenül ugyanúgy képesek a jelet feldolgozni (O/E/O átalakítással).

Érdemes lehet foglalkozni rugalmas hullámhosszosztással is.

Irodalomjegyzék

- [1] Dr. Szigeti János. „Network state advertisement and p-cycle protection for reliable connections” PhD thesis, BME TMIT 2010
- [2] Dr. Cinkler Tibor, Dr. Szigeti János. Hullámhossz kiosztási stratégia kidolgozása vegyes 10G/100G rendszerekre, BME-TMIT 2014
- [3] Biswanath Mukherjee (1997): *Optical Communication Networks*. McGraw-Hill, Inc. USA (ISBN 0-07-044435-8)
- [4] Michael Pioro, Deepankhar Medhi (2004): *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers (ISBN 0-12-557189-5)
- [5] ITU-T kiadvány DWDM alapú optikai átviteli közegű rendszerek megvalósítására http://www.itu.int/dms_pub/itu-t/oth/1D/01/T1D010000090001PDFE.pdf (2015.10.26)
- [6] ITU-T Study Group 15 – Távközlő média és optikai rendszerek karakterisztikája G.600–G.699 ajánláscsalád - <http://www.itu.int/rec/T-REC-G/en>
- [7] ITU-T G.694.1 ajánlás a spektrum felosztásáról WDM alkalmazásokhoz, DWDM frekvencia táblázat <http://www.itu.int/rec/T-REC-G.694.1-201202-I/en> (2015.10.26)
- [8] ITU-T G.652 Egymódosú optikai szálak karakterisztikája. <http://www.itu.int/rec/T-REC-G.652-200911-I/en> (2015.10.26)
- [9] Yasin, Zafar. *Optical Fiber Communication*. előadás : Stanford University. <https://www.slac.stanford.edu/slac/sass/talks/opticalfiber.pdf> (2015.10.26)
- [10] Jeff Hecht (2011): „Ultrafast fibre optics set new speed record”, *NewScientist*, 2809. kiadás <http://www.newscientist.com/article/mg21028095.500-ultrafast-fibre-optics-set-new-speed-record.html> (2015.10.26)
- [11] Fiberdyne Labs weboldala <http://www.fiberdyne.com/products/itu-grid.html> (2015.10.26)
- [12] Koaxiális kábelek paramétereit http://www.madaboutcable.com/cables/coaxial_cables/products/rg400_coaxial_cable.html (2015.10.26)
- [13] Fiberguide Industries weboldala, *2D Fiber Optic Arrays* http://fiberguide.com/wp-content/uploads/2012/08/V-Grooves_Arrays_FINAL.pdf (2015.10.26)

- [14] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo, „Elastic optical networking: a new dawn for the optical layer?,” *Communications Magazine*, IEEE, vol. 50, no. 2, pp. s12–s20, 2012.
- [15] I. Tomkos, P. S. Khodashenas, J. M. Rivas-Moscoso, D. Klouidis, „Switching and Routing for spectrally and spatially flexible optical networking.”, Athens Information Technology (AIT)
http://eprints.aston.ac.uk/25320/1/Spatial_spectral_flexible_optical_networking.pdf (2015.10.26)
- [16] IP alapú hálózatok menedzsmentje tantárgy, BME VIK VITMA365 tananyag, Teliasonea előadás, 2014
- [17] J. P. Vasseur, M. Pickavet, P. Demeester, (2004):*Network Recovery*. Morgan Kaufmann Publishers (ISBN 0-12-715051-x)