

# Távközlési hálózatok tervezése a forgalomeloszlás változásainak figyelembevételével

TAMÁSI LEVENTE, JÓZSA BALÁZS GÁBOR, ORINCSAY DÁNIEL

{Levente.Tamasi, Balazs.Jozsa, Daniel.Orincsay}@eth.ericsson.se

**Kulcsszavak:** időfüggő forgalom, forgalom-átterhelés, kihasználtságnövelés, útvonalválasztás

Jelen tanulmány egy új algoritmust javasol távközlési hálózatok költséghatékony tervezésére. Az alkalmazott hálózati modell kétféle hálózati elemet használ: útvonalválasztókat és átviteli utakat. A gyakorlatban ezek kapacitásai diszkrét értékek, ezért az egyes hálózati eszközökhöz lépcsős költségfüggvények rendelhetők. A forgalom periodikus (napi, heti) változásai is figyelembe vehetők a hálózattervezés során, ami olcsóbb költségű hálózatot eredményezhet.

Írásunkban egy új algoritmust javasolunk, amely egyszerre oldja meg a hálózattervezés és az útvonalválasztás problémáját. A bemutatott módszer kiindulásként egy hatékony, globális útvonal-optimalizáláson alapuló heurisztikus algoritmust alkalmaz, azonban kiterjeszti azt több forgalmi időszak kezelésére. Az új algoritmus teljesítményét szimulációval vizsgáljuk két referenciaalgoritmus segítségével, valós és véletlenszerűen generált problémapéldányokon.

## 1. Bevezetés

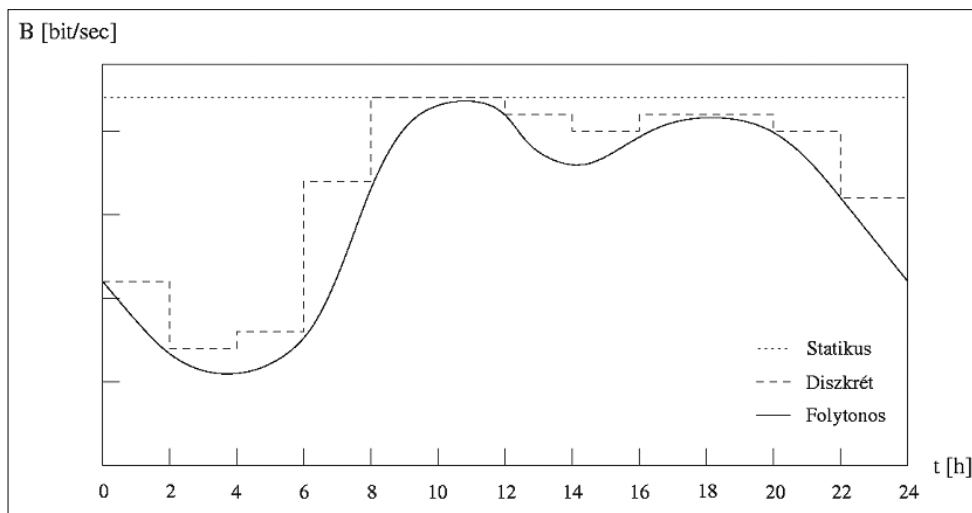
Napjainkban a sávszélesség iránti igény rohamosan növekszik, mivel egyrészt új, nagy sávszélesség-igényű alkalmazások jelennek meg, másrészt a felhasználók száma is gyorsan emelkedik. Emellett megfigyelhető az újabb és újabb technológiák elterjedése is. A fentiek miatt gyakran felmerülő probléma a kommunikációs hálózatok tervezése és méretezése. Mivel a gerinchálózatok kiépítése jelentős költségű beruházás, indokolt egy olyan algoritmus megtervezése, amely ezt a feladatot úgy oldja meg, hogy a keletkező hálózat költsége minél kisebb legyen. Ennek a célnak az eléréséhez fontos a hálózat jó kihasználtságának biztosítása, amihez célszerű figyelembe venni a forgalomeloszlás napi változásait. A jó kihasználtság a szolgáltatók számára fontos szempont, mivel így nagyobb profitra tehetnek szert. A napi forgalomeloszlás-változásokat figyelembe vevő megközelítés előnye, hogy a hálózattervezés során kihasználható, hogy az egyes útvonalválasztó-párok közötti maximális kapacitásigény fellépése külön-

böző időszakokra eshet, így alacsonyabb eszközkapacitások is elegendőek lehetnek a hálózatban a legforgalmasabb időszakokra tervezéshez képest.

Az 1. ábrán látható a folytonos forgalmi függvény statikus és diszkrét modellezése. A statikus modellezés során az útvonalválasztó-párok között a nap folyamán fellépő maximális kapacitásigényre tervezünk hálózatot, míg diszkrét modellezés esetén a napot több – akár különböző hosszúságú – időszakra osztjuk, és az egyes időszakokra külön-külön adjuk meg a forgalmi igényeket.

Jelen tanulmányban ez utóbbi megközelítést alkalmazzuk a napi forgalomeloszlás-változások modellezésére. Ehhez szükséges, hogy a forgalmi igényekhez rendelt útvonalrendszer a nap folyamán dinamikusan átkonfigurálható legyen. Ezt a lehetőséget több technológia, így például az egyre jobban terjedő MPLS (Multi-Protocol Label Switching, többprotokollos címkékapcsolás) biztosítja. A napi forgalomváltozások által indokolt újrakonfigurálások témájával az ITU-T E.360.6 [1] ajánlása is foglalkozik, és a kapacitásmenedzsment fogalma alá sorolja.

1. ábra A folytonos forgalmi függvény statikus és diszkrét modellezése



Az ajánlás több módszert is javasol a periodikus forgalomeloszlás-változásokat figyelembe vevő hálózat-tervezésre. A *DEFO* (*Discrete Event Flow Optimization*) modellek a forgalmi igényeket diszkrét hívási eseményekké alakítják, a *TLFO* (*Traffic Load Flow Optimization*) modellek az igények leírásához a lineáris programozás eszköztárát veszik igénybe, míg a *VTFO* (*Virtual Trunking Flow Optimization*) modellek az igényeket előre definiált kapacitásegységekké konvertálják. Mindhárom módszer iteratív módon oldja meg a problémát: egy inicializálási fázist követően felváltva ismétlődik az útvonaltervezés és a kapacitástervezés.

Áramkörkapcsolt hálózatokra Medhi [2] és Ouveysi [3] ad a napi forgalomeloszlás-változásokat kihasználó hálózat-tervezési módszert. Medhi [2]-ben bemutatott módszere az egyszeres meghibásodások ellen is védelmet nyújt. Kétszintű modellt alkalmaz, melynek lényege, hogy együtt kezeli a rendezőket (*cross connect*) és fizikai összeköttetéseket tartalmazó fizikai hálózat, valamint a kapcsolókból és logikai összeköttetésekből álló logikai hálózatot. A tervezés célja adott szolgáltatásminőség biztosítása a logikai hálózatban hibamentes működés, valamint egy fizikai összeköttetés meghibásodása esetén. A probléma megoldásához heurisztikus algoritmusokat javasol, melyek a probléma hibaállapotok szerinti dekompozícióján alapulnak.

Ouveysi [3]-ban javasolt algoritmus teljesen összekötött (*full mesh*) áramkörkapcsolt hálózatokra hasonló ötleten alapul. Modelljében minden kapcsolópár közötti forgalom vagy a közvetlen összeköttetésen, vagy egy harmadik kapcsoló közbeiktatásával zajlik. Megadja a probléma egy lineáris programozási megfogalmazását, és egy forgalmi időszakok szerinti dekompozíció alapuló heurisztikus algoritmust javasol.

*ATM* (*Asynchronous Transfer Mode*) hálózatokra Bauschert [4], valamint Medhi [5] [6] [7] ad módszereket. Bauschert [4]-ben javasolt algoritmusának célja adott szolgáltatásminőség garantálása és a virtuális útvonalak (*virtual path*, *VP*) rendszerének optimális kialakítása azzal a feltételezéssel, hogy egy virtuális áramkör (*virtual circuit*, *VC*) több virtuális útvonalat is igénybe vehet. Módszere a feladatot három részproblémára – virtuális áramkörök útvonaltervezése, virtuális útvonalak méretezése és virtuális útvonalak tervezése – bontja, melyeket iteratív módon old meg, két részfeladatot mindig állandónak tekintve. A forgalomeloszlás periodikus változásait a virtuális áramkörök útvonaltervezése során veszi figyelembe.

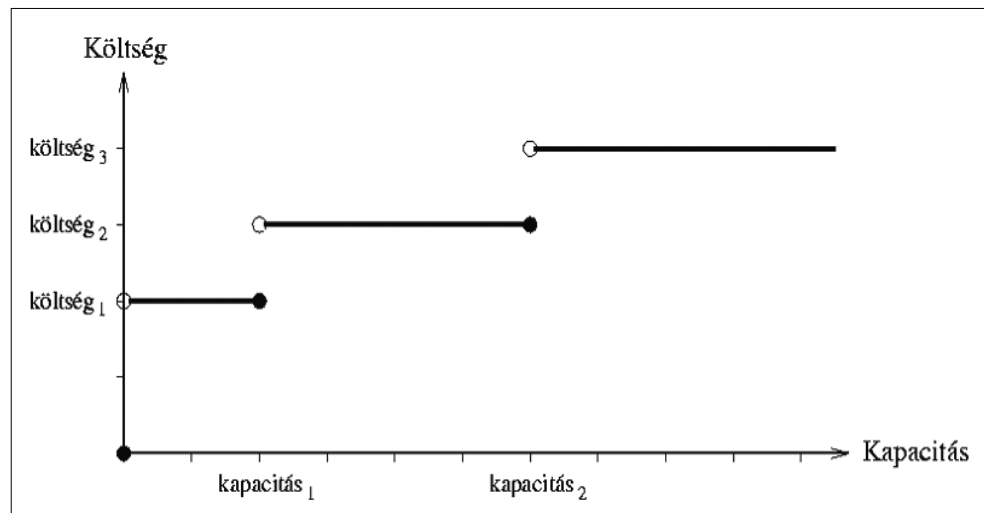
Medhi [5,6] cikkeiben bemutatott módszerei *ATM* hálózatokra eltérő modellt használnak: minden útvonalválasztó-pár esetén minden forgalmi osztályhoz kü-

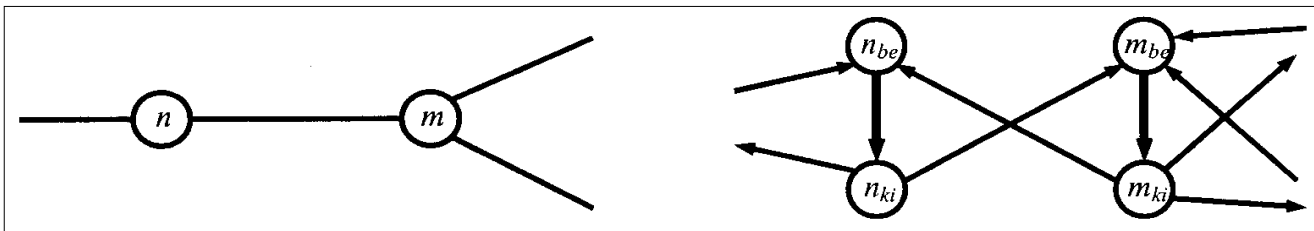
lön-külön virtuális útvonalat rendelnek. Medhi megmutatja, hogy az egyes virtuális útvonalakon belül statisztikus, a virtuális útvonalak között pedig determinisztikus nyálábolást használva a probléma két részfeladatra bomlik: egy sáv szélesség-becslési és egy kombinált útvonal- és kapacitástervezési problémára. A feladat megoldására több algoritmust is javasol: egy genetikus algoritmust, egy Lagrange-relaxáción alapuló módszert és az előző kettő ötvözetét.

Medhi és Lu [7]-ben javasolt módszere a [2]-ben bemutatotthoz hasonló modellt alkalmaz *ATM* hálózatokra azzal a különbséggel, hogy mind a virtuális útvonalak, mind a virtuális összeköttetések esetén dinamikusan változó útvonalakat tételeznek fel. Módszerük a probléma kevert egész értékű lineáris programozási megfogalmazásán alapul, melyet a két hálózati szint szerinti dekompozícióval oldanak meg.

Látható tehát, hogy a periodikus forgalomeloszlás-változások figyelembevételével történő hálózat-tervezés problémájával többen foglalkoztak az irodalomban. Azonban a fent bemutatott módszerek közös jellemzője, hogy költségfüggvény-modelljük nem közelíti megfelelően a valóságot. A gyakorlatban az eszközök moduláris felépítésűek, fix – általában szabványos – kapacitású egységekből állnak, ezért költségfüggvényük lépcsős, és csökkenő határköltséget mutatnak (2. ábra). Emellett a [2,5,6,7] cikkekben javasolt módszerek fontos tulajdonsága, hogy előre számított útvonalak közül választanak az útvonal-optimalizálás során, amely jelentősen korlátozza az algoritmus lehetőségeit. A fentiek indokolják egy hatékony, a forgalomeloszlás periodikus változásait is figyelembe vevő, lépcsős költségfüggvény modellt alkalmazó hálózat-tervező algoritmus kifejlesztését. A probléma NP-nehéz, vagyis – a tudomány mai állása szerint – nem adható rá olyan algoritmus, amely polinom időben garantálja az optimális megoldás megtalálását. Ezért indokolt a heurisztikus megközelítés alkalmazása. Az általunk javasolt algoritmus a [8]-ban bemutatott heurisztikus algoritmusból indul ki, és azt terjeszti ki több forgalmi időszak kezelésének képességére.

2. ábra Példa lépcsős költségfüggvényre





3. ábra Csúcs helyettesítése virtuális éllel

## 2. A tervezési probléma

Az alábbiakban bemutatjuk a tervezési probléma formalizálásához alkalmazott hálózati, forgalmi és költségmodell. A hálózatot egy irányított gráf segítségével reprezentáljuk, a csúcsok az útvonalválasztókat, az élek a fizikai összeköttetéseket jelképezik. A csúcsokon és éleken kapacitáskorlátokat definiálunk az egyes eszközöknek megfelelően. Mivel a gráfalgoritmusok általában élek kapacitáskorlátját tudják könnyen kezelni, a csúcsokat virtuális éllel helyettesítjük. Ezek szerint egy  $n$  csúcs helyét két új csomópontot veszi át: az eredeti csúcsba bejövő forgalom az  $n_{be}$  csomópontba érkezik, a kimenő forgalom pedig az  $n_{ki}$  csomópontból indul. Az útvonalválasztó kapacitáskorlátját az  $(n_{be}, n_{ki})$  virtuális élen vesszük figyelembe (3. ábra).

Az útvonalválasztó-párok közötti forgalmat diszkrét módon modellezzük: a nap rögzített számú időszakra oszlik, és minden egyes időszakra külön-külön adottak a forgalmi igények. Az időszakok hossza eltérő is lehet, ezért minden időszakhoz egy súlyt is rendelünk. Az alkalmazott forgalmi modell az úgynevezett pipe-modell: az egyes forgalmi igényeket a forrás és nyelő útvonalválasztó, valamint az igényelt sáv szélesség definiálja. Bár a továbbiakban egyetlen oszthatatlan forgalmi igényt feltételezünk minden útvonalválasztó-pár között, a modell képes egy útvonalválasztó-pár között több igényt is kezelni. Így lehetőség van több forgalmi osztály alkalmazására is; elkülöníthető például a hang- és az adatforgalom.

A különböző időszakokban fellépő forgalmi igények elvezetése során az igényekhez egy-egy kapacitásfoglalt útvonalat rendelünk, ami azt jelenti, hogy az elvezetéshez elegendő kapacitást kell lefoglalni az igényhez rendelt útvonalon elhelyezkedő útvonalválasztókban és fizikai összeköttetéseken. Természetesen egy adott hálózati eszközön történt kapacitásfoglalások összege nem haladhatja meg az adott eszköz teljes kapacitását.

Az egyes hálózati eszközök ár/kapacitás viszonyainak modellezésére jelen tanulmányban lépcsős költségfüggvényeket alkalmazunk, amelyek minden hálózati eszközre (útvonalválasztók, illetve fizikai összeköt-

etések) egyediek lehetnek. Ez lehetővé teszi speciális költségmódosító faktorok, például már kiépített, ingyen rendelkezésre álló hálózati eszközök figyelembe vételét is.

A fentiek tükrében a tervezési feladat a következő módon definiálható. Az algoritmus bemenetei:

- az útvonalválasztók elhelyezkedése,
- az útvonalválasztók közötti lehetséges fizikai összeköttetések,
- az egyes hálózati eszközök (útvonalválasztók, illetve fizikai összeköttetések) egyedi költségfüggvényei,
- az időszakokhoz rendelt súlyok,
- minden egyes időszakra a forgalmi igények halmaza.

A tervezési feladat egy megoldása, vagyis az algoritmus kimenete az alábbiakat tartalmazza:

- a hálózati topológiát (a lehetséges fizikai összeköttetések közül nem feltétlenül szerepel az összes a megoldásban),
- az egyes hálózati eszközök kapacitását,
- minden egyes időszakra a forgalmi igényekhez rendelt útvonalakat.

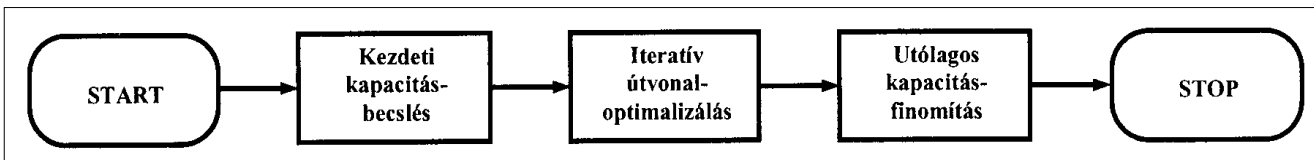
## 3. A javasolt hálózat-tervező algoritmus

Az alábbiakban először vázoljuk a javasolt algoritmus felépítését, majd részletesen is bemutatjuk az egyes fázisok működését. Az algoritmus a *Multi-Hour Core Network Designer (CND<sup>MH</sup>)* nevet kapta.

### 3.1. Az algoritmus felépítése

A forgalomeloszlás napi változásait figyelembe vevő hálózat-tervezési probléma megoldására kifejlesztett algoritmus a [8]-ban ismertetett heurisztikus algoritmusból (a továbbiakban ezt értjük *eredeti algoritmus* alatt) indul ki, amely hatékony megoldást nyújt az egy időszakkal, vagyis egy igényhalmazzal dolgozó hálózat-tervezési feladatra. Az új algoritmus három fázisból áll (4. ábra), melyek az eredeti algoritmus fázisainak kiterjesztései több forgalmi időszak kezelésére:

4. ábra Az algoritmus folyamatábrája



- *Kezdeti kapacitásbecslés (KKB)*: feladata az összes időszak forgalmi igényeinek függvényében megbecsülni a szükséges eszközkapacitásokat, hogy megfelelő kiindulási alapot biztosítson a következő fázis számára. Az eredményül kapott hálózatban tipikusan nem vezethető el az összes forgalmi igény, vagyis ez a fázis általában még nem ad megoldást a problémára.
- *Iteratív útvonal-optimalizálás (IUO)*: célja a forgalmi igények elvezetése egy globális útvonal-optimalizáló algoritmus segítségével oly módon, hogy a fázis végére minden időszakban az összes forgalmi igény kielégíthetővé váljon, ugyanakkor a hálózat összköltsége minél kisebb legyen. Ez a fázis már teljes megoldást ad az adott problémára, de az eredmény költsége a harmadik fázisban még finomítható.
- *Utólagos kapacitás-finomítás (UKF)*: megkísérli a hálózat összköltségének mérséklését azáltal, hogy megpróbálja csökkenteni azon hálózati eszközök méretét, melyekről viszonylag kicsi sáv szélességet más eszközökre terelve kisebb eszközméret is elegendő lenne.

### 3.2. Kezdeti kapacitásbecslés

A KKB fázis kezdetben minden időszakra egy két lépésből álló iterációt végez: az algoritmus először megkeveri az adott időszak forgalmi igényeit, majd a kapott sorrendben elvezeti azokat a hálózatban Dijkstra algoritmusának felhasználásával. Az elvezetés során alkalmazott súlyfüggvény előnyben részesíti az alacsony fajlagos költségű eszközöket. A kezdeti kapacitásbecslő algoritmus ezután minden eszköz kapacitását az iterációk során az adott eszközön előforduló legkisebb értékre állítja minden időszak minden elvezetését figyelembe véve. Emögött az a megfontolás áll, hogy amennyiben az összes időszak összes elvezetése során szükség volt egy bizonyos kapacitásra az adott eszközön, akkor valószínűleg optimális esetben is legalább ekkora eszközméret szükséges.

### 3.3. Iteratív útvonal-optimalizálás

Az algoritmus második (IUO) fázisa szintén egy két lépésből álló iteráció: először sorra minden időszakra megpróbálja elvezetni a forgalmi igényeket az alkalmazott globális útvonal-optimalizáló segítségével figyelembe véve a fennálló kapacitáskorlátokat. Ha ez a lépés sikeres, a fázis véget ér, ellenkező esetben pedig egy – a hálózatba be nem fért igények által meghatározott – eszköz kapacitását megnöveli, és újra megpróbálkozik az elvezetéssel. A kiválasztáshoz használt metrikának minden időszak forgalmi igényeit számításba kell vennie. Ezért a következő metrika alkalmazása mellett döntöttünk: az algoritmus minden időszakban elvezeti a hálózatba be nem fért igényeket egy kapacitáskorlátok nélküli gráfban, és az elvezetések során eszközönként összegzi, hogy a hálózatba be nem fért igények az adott eszköz kapacitáskorlátját mennyivel sértették volna, ha a korlátok figyelembe vételével történt volna az elvezetés. A második lépésben ezután

azt az eszközt választja, amelyre a fenti összegzett kapacitáskorlát-sértés a legnagyobb. Az összegzés során az algoritmus az egyes időszakok súlyát is figyelembe veszi. A módszer fontos előnye, hogy a fázis által alkalmazott globális útvonal-optimalizáló modulárisan lecserélhető bármely hasonló célú algoritmusra. Jelen tanulmányban a [9]-ben javasolt módszert alkalmazzuk.

### 3.4. Utólagos kapacitás-finomítás

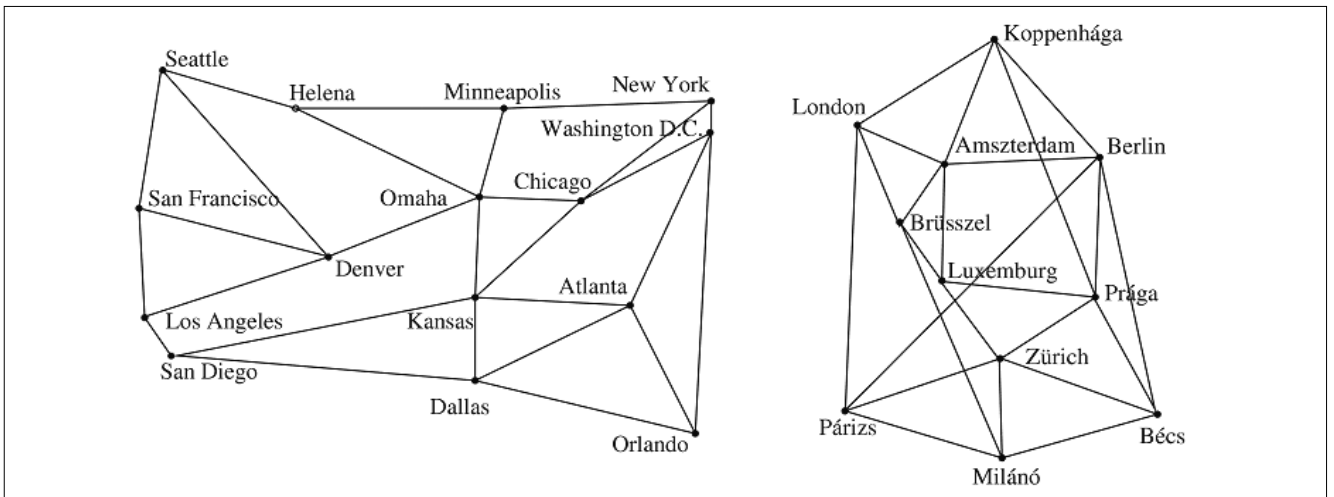
Az algoritmus harmadik fázisának célja a hálózat összköltségének finomítása oly módon, hogy megkísérli azon hálózati eszközök méretének csökkentését, amelyek gyengén kihasználtak olyan értelemben, hogy viszonylag kis forgalmat más eszközökre terelve egy kisebb méretű eszköz is elegendő lenne. Ehhez a jelen-séghez kapcsolódóan bevezetjük a *relatív lépcsőkihasználtság* mértékét, amely megadja, hogy az adott eszközt használó forgalmi igények összsáv szélessége és az eggyel kisebb kapacitáslépcsőhöz tartozó kapacitásérték különbsége hogyan aránylik az aktuális kapacitáslépcső hosszához. Az alkalmazott algoritmus először sorbarendezi az eszközöket a relatív lépcsőkihasználtságok forgalmi időszakok felett képzett súlyozott átlaga szerint. Ezután végiglépked az eszközökön, és az aktuális eszközt egy lépcsővel kisebb kapacitására állítja, majd újrafuttatja a teljes második fázist (IUO) az új kapacitáskorlátokkal. Az IUO adott esetben egyes eszközök kapacitását megnövelheti, ezért újrafuttatása után a hálózat költsége nem feltétlenül csökken. Amennyiben költségcsökkenés következik be, újraindul a harmadik fázis (UKF), ellenkező esetben az algoritmus visszatér a csökkentés előtti megoldáshoz, és a következő eszközre lép. A fázis akkor ér véget, ha egyik eszköz kapacitását sem sikerült csökkenteni. A módszer UKF fázisa során alkalmazott algoritmus a futás során kivonja a további vizsgálatok alól azokat az eszközöket, melyeket egymás után többször nem sikerült kisebb kapacitására cserélni.

## 4. A teljesítményvizsgálat eredményei

A következőkben bemutatjuk a javasolt CND<sup>MH</sup> algoritmus teljesítményvizsgálata során a tesztproblémák előállításához alkalmazott módszereket és a két referenciaalgoritmust. Ezt követően ismertetjük a vizsgálatok legfontosabb eredményeit.

### 4.1. Tesztproblémák generálása

A teljesítményvizsgálat során az algoritmust különböző – valós és véletlenszerűen generált – problémapéldányokra futtattuk. A következőkben ismertetjük a problémapéldányok előállításának részleteit, azaz a hálózati topológiák, a forgalmi igényhalmazok és a költségfüggvények generálását. A problémapéldányok előállítása a [8]-ban ismertetetthez hasonló, de a forgalom generálásakor egy bonyolultabb módszer alkalmazására volt szükség a periodikus forgalomeloszlás-változások modellezéséhez.



5. ábra A teljesítményvizsgálathoz használt két valós topológia

A véletlenszerű hálózati topológiák előállítását a [9]-beli módszerrel végeztük, amely a valós hálózatok jellemzőit jól közelíti. Jelen tanulmányban 15, 25 és 35 útvonalválasztót tartalmazó topológiákra mutatunk be eredményeket. Az útvonalválasztók kiindulási átlagos fokszáma minden esetben 5 (a kezdetben adott fizikai összeköttetések közül nem feltétlenül szerepel az összes a végső hálózatban). A véletlenszerű topológiák mellett az 5. ábrán látható két valós hálózatra (a továbbiakban USA és Európa) is vizsgáljuk az algoritmus teljesítményét.

A forgalmi igények meghatározása során a [8]-beli módszerből indultunk ki, azonban a nap folyamán változó forgalom előállításához több módosításra volt szükség. A módosított forgalomgeneráló algoritmus lépései:

- A hálózati topológia felosztása különböző területekre, úgynevezett *régiókra* úgy, hogy minden régióba megközelítőleg azonos darabszámú csomópont kerüljön. A régiókra való felosztás során az algoritmus a gráf középpontjához legközelebbi útvonalválasztókat besorolja egy központi régióba, a többi régióba pedig az azonos szögtartományba eső útvonalválasztók kerülnek (6. ábra).

- A régiók párba állítása minden időszakra. Ezzel a megoldással modellezzük azt a jelenséget, hogy a nap folyamán a hálózat más-más területei között lép fel nagyobb forgalom. A párba állítás úgy történik, hogy két régió egymásnak ne lehessen párja egynél több időszakban. A vizsgálatok során a régiók és az időszakok száma is 6 volt, az időszakokhoz pedig egységnyi súlyt rendeltünk.

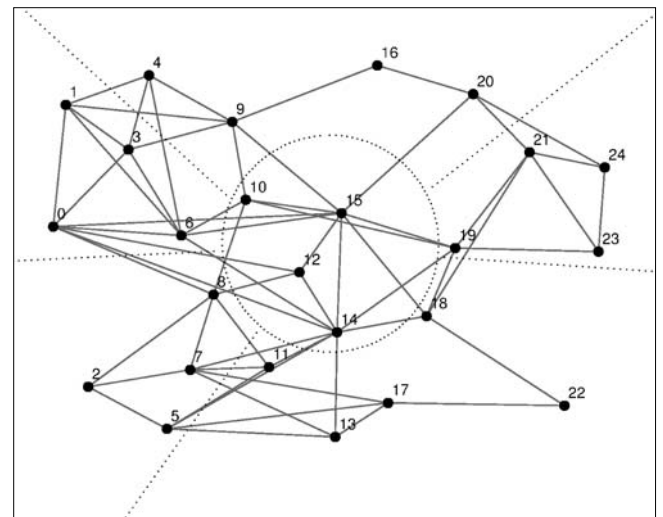
- A forgalmi igények meghatározása minden időszakra egy  $\Delta$  paraméter, az úgynevezett *régiós torzítási faktor* figyelembevételével. Egy adott időszakban a régióhoz rendelt pár kitüntetett szerepet játszik: a régióból induló forgalom egy része kijelölten a régió párjába irányul. Ennek a forgalomnak az arányát a  $\Delta$  régiós torzítási faktor adja meg. A régióból induló forgalom fennmaradó része egyenletesen oszlik meg az összes régió között, beleértve magát a régiót és az adott időszakbe-

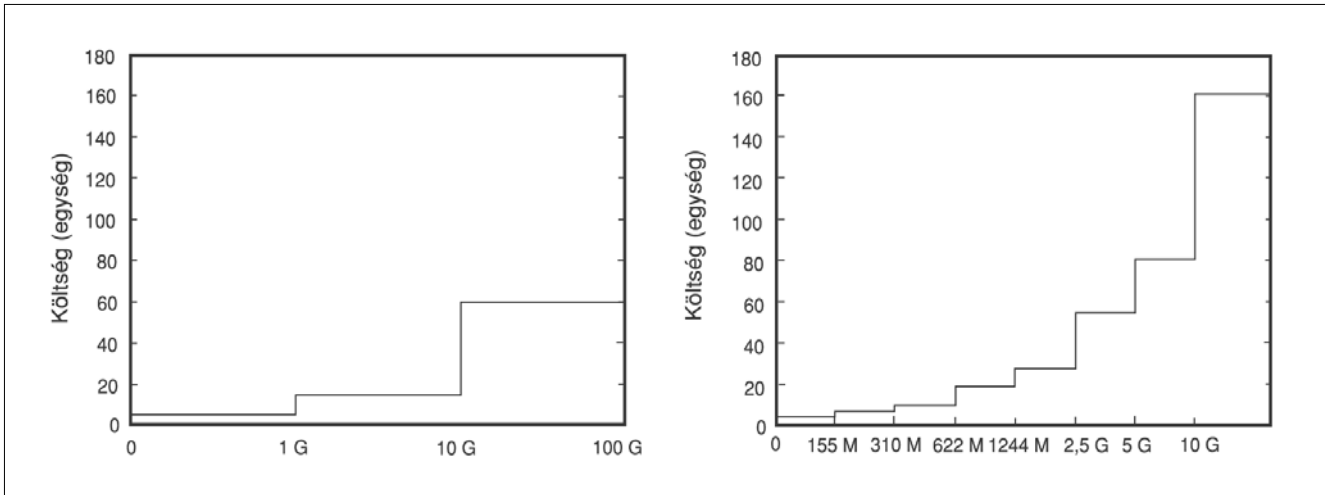
li párját is. Így a  $\Delta=0\%$ -os régiós torzítás jelenti a teljesen egyenletes forgalomeloszlást,  $\Delta=100\%$  esetén pedig a régióból induló teljes forgalom a régió párjába jut. A vizsgálatok során öt régiós torzítási faktor mellett elemeztük az algoritmusok teljesítményét: 0%, 25%, 50%, 75%, 100%.

- Az egyes időszakok forgalmi igényeinek skálázása a *forgalomvolumen* paraméternek megfelelően, amely az útvonalválasztó-párok közötti teljes forgalom nagyságát írja le. Az igények skálázása úgy történik, hogy az egyes időszakok igényhalmazának elemeit Dijkstra algoritmusával elvezetve a hálózatban egy fizikai összeköttetésre átlagosan ezen paraméter által megadott nagyságú forgalom jusson.

A költségfüggvények generálásához az alábbi módszert használtuk. Az útvonalválasztóknál három, a fizikai összeköttetésekénél kilenc kapacitásszintet definiáltunk. A fizikai összeköttetések esetén a kapacitásszintek az *STM (Synchronous Transfer Mode)* szabvány kapacitásszintjeinek felelnek meg, illetve azoknak az eseteknek, amikor párhuzamosan két azonos kapacitású eszközt telepítenek.

6. ábra A hálózati topológia régiókra osztása

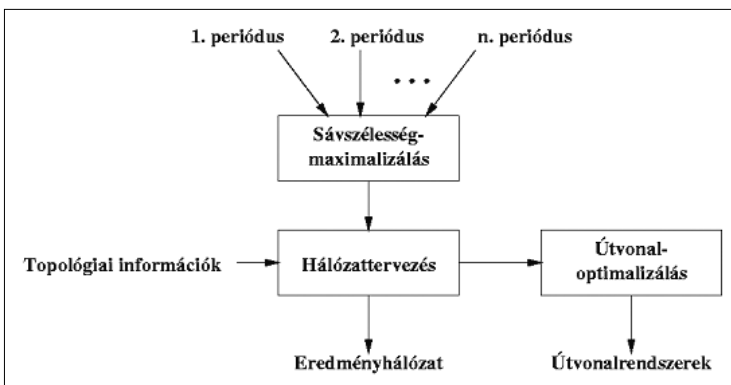




7. ábra A vizsgálatok során alkalmazott költségfüggvények

A költségfüggvény azon a feltételezésen alapul, hogy két azonos kapacitású eszköz költsége kisebb, mint az eggyel nagyobb kapacitásúé, azonban három azonos méretű eszköz telepítésénél jobban megéri eggyel nagyobb eszközt telepíteni. Emellett a 0 kapacitásszint is megengedett, ami azt jelenti, hogy az adott két útvonalválasztó között a tervezett hálózatban nincs fizikai összeköttetés, vagyis a tervezőalgorithmus az adott fizikai összeköttetést elhagyta a kiindulási halmazból. Az útvonalválasztók és fizikai összeköttetések kiindulásként használt kapacitásértékeit és a hozzájuk tartozó költségeket a 7. ábra mutatja. A kiindulási értékeket minden hálózati eszköznél véletlenszerűen torzítja az algoritmus, és így egyedi költségfüggvényeket hoz létre.

8. ábra A sávszélesség-maximalizáló (SM) referenciaalgorithmus folyamatábrája



#### 4.2. Referenciaalgorithmusok

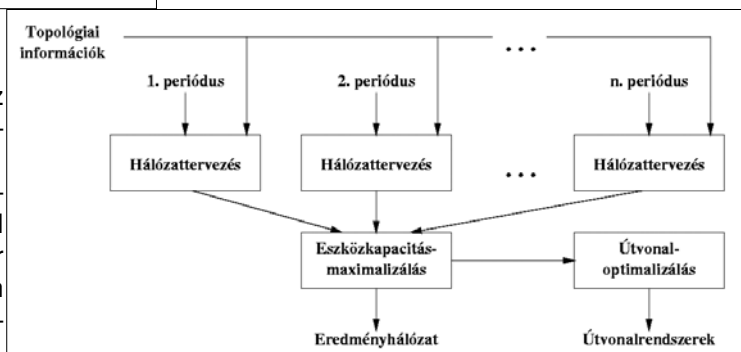
Az algoritmus teljesítményének vizsgálatához az eredeti algorithmusból kiindulva két referencia-módszert implementáltunk.

A sávszélesség-maximalizáló (SM) referenciaalgorithmus a statikus forgalommodellezésnek felel meg. Az algoritmus minden útvonalválasztó-pár esetén meghatározza a közöttük a nap folyamán fellépő maximális kapacitásigényt, és erre a maximalizált igényhalmazra futtatja le az eredeti algo-

ritmust az egyes hálózati eszközök kapacitásának kiszámításához (8. ábra). Ezután az IUO fázis által használt globális útvonal-optimalizáló algorithmus segítségével minden időszak forgalmi igényeit elvezeti a hálózatban, hogy meghatározza az egyes időszakokhoz tartozó útvonalrendszert. Az SM referenciaalgorithmus előnye, hogy az algoritmus által az adott útvonalválasztó-pár közötti maximális forgalomra kiszámított útvonal statikus útvonalként is alkalmazható az adott útvonalválasztó-pár között különböző időszakokban fellépő forgalmi igényekre.

Az eszközkapacitás-maximalizáló (EKM) referenciaalgorithmus minden egyes forgalmi igényhalmazra külön-külön tervez egy hálózatot a kiindulási algorithmus segítségével, majd minden hálózati eszköz kapacitását az egyes időszakokra adódó hálózatokban az adott eszközön előforduló legnagyobb értékre állítja (9. ábra). Ezután az SM referenciaalgorithmushoz hasonlóan határozza meg a végleges útvonalrendszereket, vagyis a végső kapacitáskorlátok mellett az alkalmazott globális útvonal-optimalizáló algorithmus segítségével elvezeti az egyes időszakok forgalmi igényeit. Természetesen az egyes időszakokra végzett hálózat tervezés során előálló útvonalrendszerek is alkalmazhatók, de a fenti módszer hatékonyabb elvezetést tesz lehetővé.

9. ábra Az eszközkapacitás-maximalizáló (EKM) referenciaalgorithmus folyamatábrája



### 4.3. Numerikus eredmények

Ez a fejezet a szimulációs vizsgálatok eredményeit mutatja be. A véletlenszerű hálózati topológiák esetén minden hálózatméretre 5 topológiára és topológiánként 3 forgalmi szituációra, a valós topológiák esetén pedig 15 forgalmi szituációra futtattuk az algoritmusokat. A forgalomvolumen paramétert 200 és 5500 Mbit/s között változtattuk a vizsgálatok során. A legfontosabb mérték a teljesítményvizsgálat során a hálózat összköltsége volt, de emellett vizsgáltuk a futási időket is.

#### A hálózat összköltsége

Az 1. táblázat mutatja az egyes algoritmusok által tervezett hálózatok átlagos összköltségének arányait (az adott problémaosztályban a legjobb megoldást tekintve 100%-nak) az összes vizsgált forgalomvolumenre átlagolva a 15, 25 és 35 útvonalválasztót tartalmazó véletlenszerű topológiákra, valamint a vizsgált két valós hálózatra. A véletlenszerű topológiák esetén három dolgot fontos megjegyezni: először is a javasolt algoritmus az esetek túlnyomó többségében a legalacsonyabb összköltségű hálózatot eredményezte. Kivételes esetek  $\Delta=0\%$  régiós torzítási faktor mellett fordultak elő, azonban teljesen egyenletes forgalomeloszlás esetén nem is várható az eredeti algoritmusnál jobb teljesítmény. Másodszor, a régiós torzítás növelésével egy adott hálózatméret mellett a javasolt algoritmus előnye növekszik a referenciaalgoritmusokkal szemben, és a  $\Delta=100\%$  esetben elérheti a 28%-ot az EKM-mel szemben, illetve az 54%-ot az SM-mel szemben.

Harmadszor, a  $\Delta=100\%$  esetet leszámítva a hálózat méretének növelésével a javasolt algoritmus előnye enyhén csökkenő tendenciát mutat.

A valós hálózatok esetén is hasonló eredmények figyelhetők meg. A vizsgálatok azt mutatják, hogy a javasolt algoritmus az USA topológia esetén 8-34%-kal alacsonyabb költséget eredményez, mint az SM-é, és 18-24%-kal alacsonyabbat, mint az EKM-é. Az Európa topológia esetén az algoritmus alkalmazásával elérhető nyereség 7-30% az SM-hez képest, illetve 15-28% az EKM-hez képest.

#### Futási idő

A javasolt algoritmus és a referenciaalgoritmusok hozzávetőleges futási idejét a 2. táblázat tartalmazza. A szimulációk egy 450 MHz-es Ultra II processzorral és 1 GB RAM-mal rendelkező Sun Ultra Enterprise 420R számítógépen futottak. A táblázat a javasolt algoritmusra és a referenciákra az adott hálózatméret és  $\Delta$  régiós torzítás melletti minimális és maximális futási időt tartalmazza percekre kerekítve.

Látható, hogy 15 útvonalválasztót tartalmazó hálózati topológia esetén az algoritmusok futási ideje nagyon alacsony, maximum pár perc. 25 útvonalválasztó esetén a javasolt algoritmus és az EKM futási ideje a 2-20 perc, az SM-é pedig a 0-8 perc nagyságrendbe esik. 35 útvonalválasztó esetén az órás nagyságrendbe esik a javasolt algoritmus futási ideje, ami elfogadható a hálózattervezési probléma jellegét és a gyakorlatban előforduló hálózatméreteket tekintve.

Hálózat	Algoritmus	$\Delta=0\%$	$\Delta=25\%$	$\Delta=50\%$	$\Delta=75\%$	$\Delta=100\%$
15 csomópont	CND <sup>MH</sup>	100,00%	100,00%	100,00%	100,00%	100,00%
	SM	100,17%	107,32%	123,51%	145,96%	153,75%
	EKM	105,75%	108,59%	114,42%	125,36%	127,90%
25 csomópont	CND <sup>MH</sup>	100,70%	100,00%	100,00%	100,00%	100,00%
	SM	100,00%	103,97%	118,83%	139,26%	141,90%
	EKM	106,48%	106,89%	108,34%	113,99%	117,67%
35 csomópont	CND <sup>MH</sup>	104,70%	100,00%	100,00%	100,00%	100,00%
	SM	100,00%	102,39%	116,50%	134,20%	149,62%
	EKM	107,99%	107,09%	107,56%	111,44%	121,88%
USA	CND <sup>MH</sup>	100,00%	100,00%	100,00%	100,00%	100,00%
	SM	108,08%	109,14%	115,12%	126,66%	134,96%
	EKM	121,11%	121,98%	123,12%	123,69%	117,57%
Európa	CND <sup>MH</sup>	100,00%	100,00%	100,00%	100,00%	100,00%
	SM	111,42%	106,86%	120,05%	129,85%	128,74%
	EKM	126,74%	122,54%	127,83%	128,40%	115,19%

1. táblázat  
A hálózat relatív összköltsége a 15, 25 és 35 útvonalválasztós véletlenszerű, valamint a vizsgált két valós topológia esetén

Hálózatméret	Algoritmus	$\Delta=0\%$	$\Delta=25\%$	$\Delta=50\%$	$\Delta=75\%$	$\Delta=100\%$
15	CND <sup>MH</sup>	0-2	0-3	0-3	0-2	0
	SM	0	0-1	0-1	0-1	0
	EKM	0-2	0-2	0-2	0-2	0
25	CND <sup>MH</sup>	2-18	2-13	3-19	4-20	0-2
	SM	0-4	0-8	0-7	0-7	0-7
	EKM	3-24	3-26	4-23	5-16	0-1
35	CND <sup>MH</sup>	12-45	17-40	19-46	18-61	2-5
	SM	4-14	4-17	6-48	5-29	5-17
	EKM	19-49	20-189	21-60	21-66	1-3

2. táblázat  
Az algoritmusok futási ideje percben 15, 25 és 35 útvonalválasztós topológiák esetén (min-max)

25 és 35 útvonalválasztós topológiák esetén megfigyelhető, hogy a javasolt algoritmus és az EKM referenciaalgoritmus futási ideje az SM futási idejének többszöröse  $\Delta=0\%$ ,  $25\%$ ,  $50\%$ ,  $75\%$  esetén.

Ennek magyarázata, hogy az SM egyetlen maximalizált igényhalmazzal dolgozik, vagyis futási ideje az eredeti algoritmus futási idejének nagyságrendjében van. Az EKM referenciaalgoritmus minden időszak forgalmi igényhalmazára külön futtatja a kiindulási algoritmust, a javasolt algoritmus pedig szintén több igényhalmazzal számol egyszerre, ami indokolja a többszörös futási időt. Kivételt képez a  $\Delta=100\%$  régiós torzítási faktor esete: ekkor mind a javasolt algoritmus, mind az EKM referenciaalgoritmus jóval rövidebb idő alatt szolgáltat megoldást, mint az SM.

Ennek az az oka, hogy a  $\Delta=100\%$  érték miatt a javasolt algoritmusnak és az EKM-nek időszakonként jóval kevesebb forgalmi igénnyel kell számolnia (mivel minden útvonalválasztóból csak a pár-régióbeli útvonalválasztókba irányul forgalmi igény), mint a maximalizálás miatt ilyenkor is egy „teljes” (vagyis minden útvonalválasztó-pár közötti igényt tartalmazó) igényhalmazzal dolgozó SM-nek.

## 5. Összegzés és konklúzió

Jelen tanulmány a kommunikációs hálózatok napi forgalomváltozásokat is figyelembe vevő tervezését tárgyalta. Az egyes hálózati eszközök költség/kapacitás viszonyait lépcsős költségfüggvényekkel írtuk le, emiatt a vizsgált probléma az NP-nehéz kategóriába tartozik. A forgalomeloszlás napi változásait diszkrét módon modelleztük: a nap ekkor rögzített számú időszakokra oszlik, és minden időszakra külön-külön adott a forgalmi igények halmaza.

A javasolt algoritmus az összes forgalmi igényhalmazt egyszerre figyelembe véve oldja meg a hálózat-tervezési problémát. Referenciának két módszert implementáltunk: az SM minden útvonalválasztó-pár közötti napi maximális forgalomra tervez, az EKM pedig külön tervez hálózatot az egyes időszakokra, majd végül az adódó kapacitások maximumát képi minden eszközre.

A javasolt algoritmus teljesítményvizsgálatát szimuláció segítségével végeztük különböző problémapéldányokra. Az alkalmazott hálózati topológiák között valós és véletlenszerű topológiák is megtalálhatók voltak. Bemutattuk, hogy a javasolt algoritmus segítségével jelentős költségcsökkenés érhető el mind az SM, mind az EKM referenciaalgoritmushoz képest. A javasolt algoritmus futási ideje még a legnagyobb vizsgált hálózatmérték esetén is elfogadható volt.

A tárgyalt problémával kapcsolatban további vizsgálatok lehetségesek. A javasolt algoritmus esetében különböző fejlesztések képzelhetők el, ilyen például a védelmi útvonalak alkalmazásának lehetősége. Emellett a forgalmi igényhalmazok előállításakor más régiógenerálási és régiópárosítási módszerek hatását is érde-

mes megvizsgálni. A teljesítményvizsgálathoz újabb referenciamódszereket is célszerű lenne implementálni: ehhez adott esetben a probléma definíciók illesztése is szükséges lehet, például ha az adott módszer nem lépcsős költségfüggvényeket alkalmaz.

## Irodalom

- [1] ITU-T Recommendation E.360.6, QoS Routing and Related Traffic Engineering Methods – Capacity Management Methods, 2002.
- [2] Medhi, D., A Unified Approach to Network Survivability for Teletraffic Networks: Models, Algorithms and Analysis, IEEE Transactions on Communications, Vol. 42, pp.534–548, 1994.
- [3] Ouveysi, I., Network Design for Multi-Hour Traffic Profile, Proceedings of Australian Telecommunication Networks and Applications Conference (ATNAC), 1995.
- [4] Bauschert, T., Multihour Design of Multi-Hop Virtual Path based Wide-Area ATM Networks, Proceedings of the 15th International Teletraffic Congress (ITC-15), Washington, D.C., USA, 1997.
- [5] Medhi, D., Multi-Hour, Multi-Traffic Class Network Design for Virtual Path-based Dynamically Reconfigurable Wide-Area ATM Networks, Proceedings of the IEEE INFOCOM '95, Boston, MA, USA, pp.900–907, 1995.
- [6] Medhi, D., Some Approaches to Solving a Multi-Hour Broadband Network Capacity Design Problem with Single-Path Routing, Telecommunication Systems, Vol. 13, pp.269–291, 2000.
- [7] Medhi, D., Lu, C. T., Dimensioning and Computational Results for Wide-Area Broadband Networks with Two-level Dynamic Routing, IEICE Transactions on Communications, Vol. E80-B, No.2, pp.273–281, 1997.
- [8] Orincsay, D., Józsa, B. G., Távközlési hálózatok költségghatékony tervezése, Híradástechnika, Vol. 58, No.4, pp. 39–45, 2003.
- [9] Józsa, B. G., Király, Z., Magyar, G., Szentesi, Á., An Efficient Algorithm for Global Path Optimization in MPLS Networks, Optimization and Engineering, Vol. 2, No.3, pp.321–347, 2001.