TARTALOM ◆ CONTENTS

Hungarian translation of the paper "Shortest Pair of Physically-Disjoint Paths in Telecommunication Fiber Networks" in the Hungarian Telecommunications Journal

E SZÁMUNK SZERZŐI ◆ AUTHORS OF THIS ISSUE

Dr. Lajtha György felelős szerkesztő; Kenesi Zsolt, Molnár Sándor, Vidács Attila, Pozsgai Péter, Rétfalvi Ákos egyetemi hallgatók (BME); Péter Gábor távközlési szakmérnök (COMEX Kft.); Horváth Jenő rendszermérnök (Matáv Üzemviteli Igazgatóság); Olcsváry László villamosmérnök (Alcatel AHT Kft); Oravecz Zsolt villamosmérnök-közgazdász (Alcatel AHT Kft.); Beliczay Katalin villamosmérnök (Siemens Rt.), Jankó Árpád doktorandusz (BME); Soltész János project menedzser; Tomasov Pavol; Ramesh Bhandari (AT&T Bell Laboratories, New Jersey), Lencsés Ferenc villamosmérnök (COMEX Kft.); dr. Takács György; Kauser Alajos elnokhelyettes (Hírközlési Főfelügyelet); dr. Gosztony Géza főmunkatárs (Matáv Rt.); Bilszky László tudományos főmunkatárs (PKI-FI).

Dr. György Lajtha senior editor, Zsolt Kenesei, Sándor Molnar, Attila Vidács, Peter Pozsgai, Akos Rétfalvi (Budapest Technical University); Gabor Péter telecommunications engineer (COMEX Ltd); Jenő Horvath system engineer (Matav Rt); László Olcsváry electric engineer (Alcatel AHT Ltd); Zsolt Oravecz electric engineer-economist (Alcatel AHT Ltd.), Katalin Beliczay electric engineer (Siemens Rt). Arpad Janko doctorand (Budapest Technical University), János Soltesz project manager; Tomasov Pavol, Ramesh Bhandari (AT&T Bell Laboratories, New Jersey), Ferenc Lencses electric engineer (COMEX Rt), dr. György Takacs, Alajos Kauser vice president (HIF); dr. Géza Gosztony senior coordinator (Matáv Rt); Laszlo Bilszky senior member (PKI Telecommunications Development Institute)

Optimális fizikailag független útvonalak

Azok az algoritmusok, amelyeket legrövidebb fizikailag független érpár meghatározására megadunk, két útvonal-megosztási topológiát vesznek figyelembe. A fizikailag független útvonalak megnövelik az adott hálózat megbízhatóságát, ugyanakkor az optimalizálás értelemszerűen tartalmazza a hálózat árának csökkenését.

A fényvezetők alkalmazásával a távközlési hálózatokban a megbízhatóság előtérbe kerül, ugyanis a nagy sávszélességű fényvezetős összeköttetések az eddigieknél nagyobb mennyiségű adatot szállítanak, ezért kiesésük nagy forgalom elvesztését jelenti. A szolgáltatások megbízhatósága a fizikai megosztással növelhető, azaz két fizikailag független útvonalon (csomópont-független és vezeték-független) bocsátjuk át a forgalmat. Vagyis, amikor az egyik útvonal kiesik a vezeték elvágása vagy a csomópont (központ) meghibásodása miatt, akkor a forgalmat másik, fizikailag független útvonalra lehet irányítani. Egyértelmű, hogy amikor ilyen fizikailag független útvonal-párok állnak rendelkezésre, akkor ajánlatos a legrövidebb érpárat választani a rendelkezésre álló párok közül (a két útvonal összege minimális), mivel ez jelenti a legalacsonyabb költséget.

Az optimális fizikailag független útvonalak algoritmusát először Suurballe [1.2] határozta meg. Mindazonáltal az algoritmusai csak olyan hagyományos gráfelméleti hálózatokra alkalmasak, amelyeket olyan (logikai) összeköttetések írnak le, amelyeket az *1. ábra* szaggatott vonalai jelölnek.

jelenti. A vezeték megosztási lehetőségek következtében, a fizikai függetlenséghez új algoritmusokra van szükség.

Az *A* és *B* csomópontok közötti tényleges útvonal tartalmazhat egy nyomvonalat vagy azok halmazát. Gazdasági és gyakorlati megfontolások szerint a fényvezetős hálózatokat úgy lehet felépíteni, hogy két különböző nyomvonal között a forgalmat megoszthatjuk. Cikkünkben olyan hálózatokkal foglalkozunk, amelyek kétféle általános módon térnek el a hagyományos, csomópontokból és összeköttetésekből álló hálózatoktól, és olyan algoritmust adunk meg, amely meghatározza a legrövidebb fizikailag független útvonalat ezen hálózat két adott csomópontja között. Tudomásunk szerint ilyen algoritmust eddig még nem adtak közre.

A HÁLÓZAT LEÍRÁSA

Az alábbi hálózat topológiákat alkalmazzuk, amelyek eltérnek a hagyományos hálózattól:

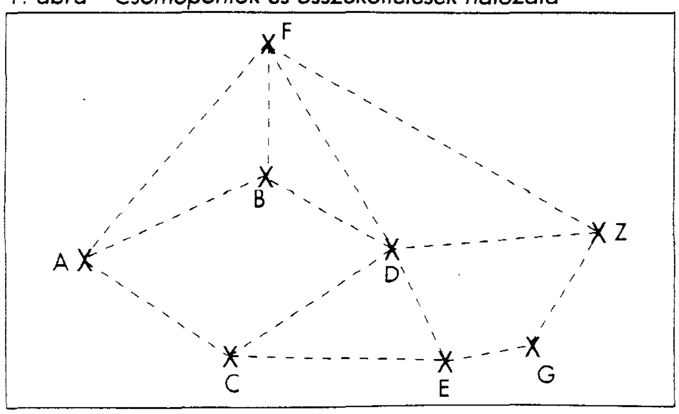
Villa konfiguráció (*2/a. ábra*):

Az *A* és *B* csomópontok fizikailag össze vannak kötve az *AO* és *OB* nyomvonalon, ahol *O* elágazási pont. Az *A* és *B* közötti összeköttetést a szaggatott *AB* vonal jelzi. Hasonló módon fizikailag az *AO* és *OC* kapcsolattal valósítjuk meg az *A* és *C* közötti összeköttetést. A *B* és *C* csomópontok nem ugyanolyan módon vannak kapcsolatban, ugyanis a *C* csomópontot a *B* csomópontból csak a *BO*, *OA*, *AO* és *OC* vezetékeken keresztül lehet elérni vagy alternatív módon (a hálózat) valamely más útvonalán, amely nem tartalmazza az *O* elágazási pontot. A *2/b. ábra* a *2/a. ábra* általánosítása. Egy *n* elágazású konfigurációt mutat, ugyanazon megkötések érvényesek a bármely B_i, B_j párra, ahol $i \neq j$; $i = 1, 2, \ldots n$ és $j = 1, 2 \ldots n$.

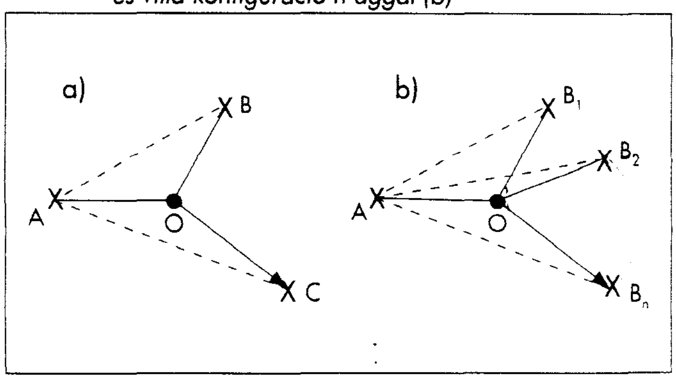
Közvetlen összeköttetés (*3. ábra*):

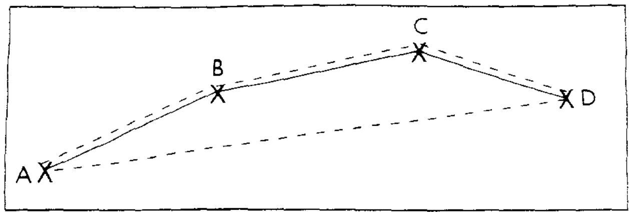
Az *AB*, *BC* és *CD* kapcsolatok (szaggatott vonalak) az *AB*, *BC* és *CD* fizikai nyomvonalon haladnak (folyamatos vonal).

1. ábra Csomópontok és összeköttetések hálózata

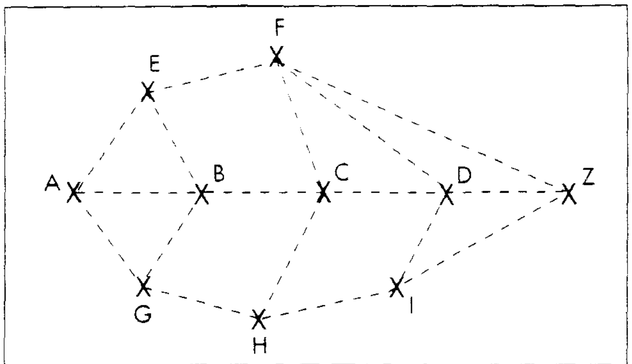


2. ábra Villa konfiguráció két ággal (a) és villa konfiguráció n ággal (b)

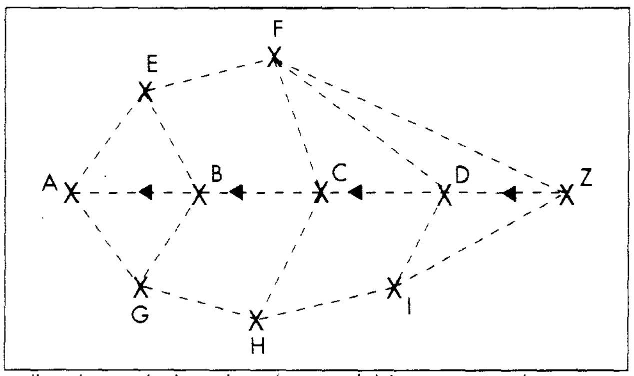




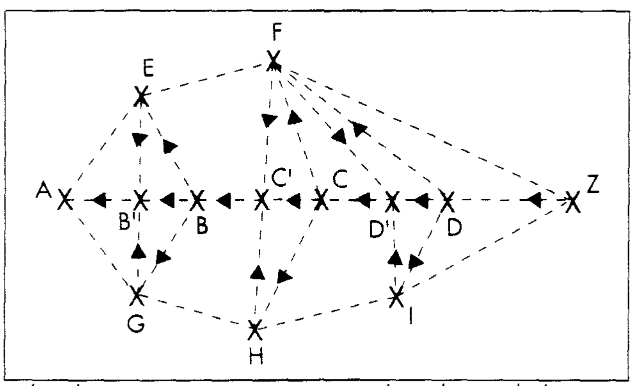
3. ábra Közvetlen AD összeköttetés, amely az AB, BC és CD összeköttetéseken keresztül vezet



5/a. ábra Tegyük fel, hogy az A és Z nyomvonal legrövidebb útvonala ABCDZ, ahol A a legrövidebb nyomvonal algoritmus kiinduló csomópontja

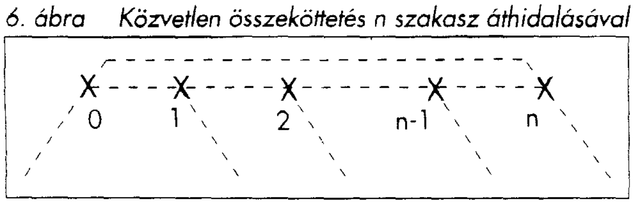


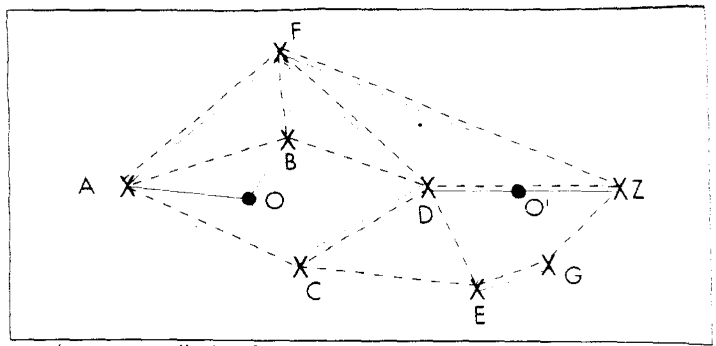
5/b. ábra A hálózatban legrövidebb nyomvonal szakaszok be vannak helyettesítve az A csomópont felé mutató negatív élekkel



5/c. ábra Csomópont megosztással módosított hálózat

6. ábra Közvetlen összeköttetés n szakasz áthidalásával





4. ábra Két villa konfigurációval rendelkező hálózat (O és O' csatlakozási pontok) és CE kifejezett összeköttetéssel (CD, DO' és O'E kombináció)

Ezenkívül az *A* és *D* csomópontok közvetlenül is össze vannak kötve egy olyan fényvezetővel, amely az *A* csomópontból indul és a *D* csomóponton végződik, továbbá keresztül megy az *AB*, *BC* és *CD* vezetékeken. Ezt a csatlakozást vagy összeköttetést a szaggatott vonal jelzi és közvetlen összeköttetésnek nevezzük. Általánosságban a közvetlen összeköttetés *n* összeköttetésen mehet át, ahol *n* egész szám és nagyobb, mint 1; továbbá az *n* összeköttetés halmaza tartalmazhatja a villa konfiguráció összekötéseit is.

A *4. ábrán* kombináljuk a 2. és 3. ábrákon bemutatott konfigurációkat.

Ez a hálózat összeköttetés szinten azonos az 1. ábra hagyományos gráf hálózatával. Fizikai szinten a hálózat különbözik attól. Két villa konfiguráció van a csatlakozási pontokon: *O* és *O'*, és egy kifejezett összeköttetés: *CE*. A probléma az, hogy a hálózatban találni kell egy optimális, fizikailag független útvonalat az adott csomópont pár között. Az optimalitás az útvonal pár minimális vezetékhosszára vonatkozik.

Világos, hogy a jelenleg rendelkezésre álló csomópont-független útvonal algoritmus [1,2], amely az 1. ábra szerinti gráfelméleti hálózatra érvényes, nem alkalmazható a 4. ábra hálózatára, mivel az *O* és *O'* csatlakozási csomópontok nem igazi csomópontok (2 ábra). Másrészről a hálózat összeköttetési szinten (szaggatott vonalak) két csomópontból áll, és a csomópont-független útvonal algoritmus összeköttetési szinten alkalmazható a 4. ábra hálózatára. Az összeköttetéseket súlyozni lehet azon vezetékek hosszával, amelyeket az összeköttetés magába foglal. Amennyiben az SPNP *(Sorter Path Node-disjointed Planning)* algoritmust összeköttetés szinten alkalmazzuk a 4. ábrára, megadja a legrövidebb útvonal-párt, (a vezeték-hosszakra vonatkozólag), ahol mindegyik útvonalat az összeköttetések egy sorozatával definiáljuk.

A megtalált két útvonal összeköttetései átfedő vezetékeket is tartalmazhatnak, fizikailag nem szükségképpen függetlenek. Például a 4. ábrán az SPNP algoritmus az *A* és *Z* csomópontok közötti optimális, fizikailag független útvonalat az *ABDZ* és *ACEGZ* pontokon át vezeti, de az *AO* és *DO'* vezetékek közösek a két megtalált útvonalra. Ezen közös vezetékek elkerülése céljából új algoritmusra van szükség. A következő fejezetben nemcsak olyan algoritmust adunk, amely fizikailag független útvonalakat ad meg (csomópont-független és egyben vezeték-független), továbbá azt is igazoljuk, hogy a meghatározott párok a legrövidebb vezetékhosszt eredményezik.

A FIZIKAILAG FÜGGETLEN ÚTVONAL PÁROK ALGORITMUSA

A *4. ábra* szerinti hálózatok optimális, fizikailag független útvonalpár meghatározására szolgáló algoritmus megkereséséhez olyan hálózat transzformációkat hajtunk végre, hogy az SPNP algoritmust tudjuk használni összeköttetés szinten, és ne sértsük meg a fizikai függetlenség követelményét. Az alábbiakban megadott SPNP algoritmus [3] a Suurballe algoritmus [1,2] továbbfejlesztett változata. Az ehhez kapcsolódó Dijkstra algoritmus kissé módosult, hogy meg lehessen kerülni a szokásos kanonikus transzformációt.

A csomópont-független útvonalak legrövidebb párjának algoritmusa [3]

① Keressük a vizsgált csomópont párhoz a legrövidebb utvonalat. Példaképpen tekintsük az *5 a.* ábrát. Mindegyik összeköttetés egyenértékű két ellentétesen irányított éllel, amelynek hossza egyenlő az összeköttetés hosszával.
② Helyettesítsük a legrövidebb nyomvonal valamennyi összeköttetését egy olyan ívvel, amely a kiindulási csomópont felé irányul *(5/b. ábra)*. Tegyük az élek hosszát negatívvá.
③ Osszuk fel a nyomvonal mindegyik csomópontját (az A és Z végponti csomópontok és a harmadfokú vagy alacsonyabb rendű [7] csomópontok kivételével) két egymás mellett elhelyezett alcsomópontra. Kössük össze ezeket a csomópontokat egy zérus hosszúságú éllel, és irányítsuk azokat a kezdő csomópont felé. Helyettesítsük a legrövidebb nyomvonal külső összeköttetéseit két éllel ugyanannak és az eredeti hosszúságnak megfelelően, és csatlakoztassuk a két alcsomóponthoz az *5/c. ábra* szerint.
④ Hajtsuk végre még egyszer a legrövidebb nyomvonal algoritmust.
⑤ Távolítsuk el a zérus hosszúságú éleket, egyesítsük az alcsomópontokat a szülő csomópontokkal. Váltsuk ki a legrövidebb nyomvonal egyszeres éleit az eredeti összeköttetésekkel (pozitív hosszakkal).

Távolítsuk el a két nyomvonal átfedő összeköttetéseit, hogy megkapjuk a legrövidebb nyomvonalat. Ezután vizsgáljuk meg a közvetlen összeköttetések és villa konfigurációk kombinált hálózatát (4. ábra).

Közvetlen összeköttetés

A *6. ábra* n szakaszból álló közvetlen összeköttetést mutat. Ahogy a 4. ábra szerinti közvetlen összeköttetés és a 0 és n csomópontok közti szakaszok nyitva hagyják annak lehetőségét, hogy a közös nyomvonalon SPNP algoritmussal meghatározott forgalom megosztás legyen. Az egyik útvonal például beléphet a 0. csomópontba és kiléphet az n. csomópontból. A másik útvonal beléphet az 1. csomópontba és kiléphet a 2, 3,...n-1 közbenső csomópontokon, amivel átfedést hoz létre az 1,...n-2 csomópontok között, miközben az útvonalnyomvonal optimalitását fenntartja az alábbi érvek szerint:

Feltételezzük, hogy I belép a 0 csomópontban. Akkor az 1, 2...n csomópontok bármelyikén kiléphet. Összesen n lehetőség van, amit két típusba lehet osztani:

a) Amennyiben I az útvonalból az 1, 2,...n-1 közbenső csomópontokon lép ki, akkor a közvetlen összeköttetés jelenléte redundáns.

b) Amennyiben I az n csomóponton lép ki, két választási lehetőség van:
- I egyetlen lépéssel közvetlen összeköttetésen az n csomópontba megy,
- I keresztül megy a 2, 3,...n-1 csomóponton és az n-edik csomópontot az utolsó szakasza végén éri el.

Mind a két választásnál azonos a nyomvonal, következésképpen a vezetékhossz vonatkozásában egyenértékűek. Az ugrások számának szempontjából az első választás ajánlatos, mivel nincs közbenső csomópont. Ennek a legrövidebb nyomvonal algoritmussal történő tervezése azonban fenntartja azzal a veszéllyel jár, hogy átfedő összeköttetések keletkeznek, amikor a másik útvonalat határozzuk meg. Következésképpen bármilyen algoritmust használunk, azt a fenti szituációban a második esetre kell korlátozni. Ez a korlátozás egyszerűen megvalósítható, ha kitöröljük a közvetlen összeköttetést a hálózatból.

a)-t és *b)* második lehetőségét vizsgálva az alábbi fontos szabály jelenik meg: Töröljük az összes közvetlen összeköttetést mielőtt az SPNP algoritmust futtatjuk.

Villa konfiguráció

Feltételezések:

① A villa konfiguráció két ágból áll (ezt a feltételezést csak a tárgyalás egyszerűsítése érdekében tettük) *(7 a. ábra)*. Ezt a konfigurációt Y konfigurációnak nevezzük. A későbbiekben látni fogjuk, az Y konfigurációra megkapott eredmények érvényesek lesznek a két ágnál többet tartalmazó elrendezésekre is.

② Nincsenek többszörös összeköttetések. Amennyiben mégis többszörös összeköttetések jelennek meg, mint a 7/a. ábrán, akkor azokat egy másodfokú vak csomópont bevezetésével ki lehet küszöbölni. *(lásd 7/b. ábra)*

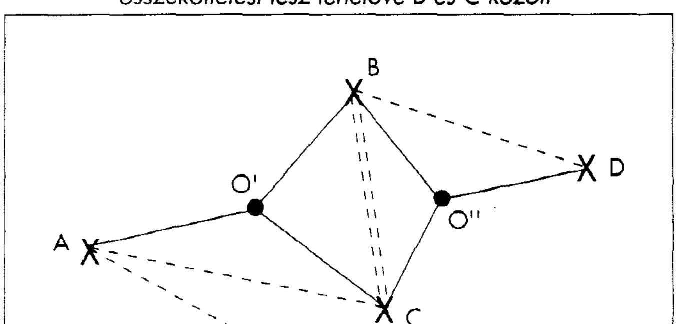
Tekintsük meg a *8. ábrát*, amely két ág találkozásával képzett Y konfigurációt mutat. Az Y konfiguráció három helyzetét vizsgáljuk [8]. A csomópont-függetlenség követelménye mindegyik esetben összeköttetés szinten biztosítja, hogy nem fog mind a két útvonal egyszerre áthaladni az *OD* összeköttetésen, amely közös a *CD*-re és a *BD*-re. Ennek oka, hogy az *OD* útvonal megosztásához szükséges, hogy a két útvonal találkozzon a *D* csomópontban. Mivel a csomópont-függetlenség kritériuma megakadályozza, hogy a két útvonalnak közös csomópontja legyen, a két útvonal nem találkozhat a *D* csomópontban, következésképpen az *OD* megosztása nem léphet fel.

Két eset fordulhat elő az A és Z (5. ábra) végpontokon, amelyek között fizikailag független útvonalakat keresünk:

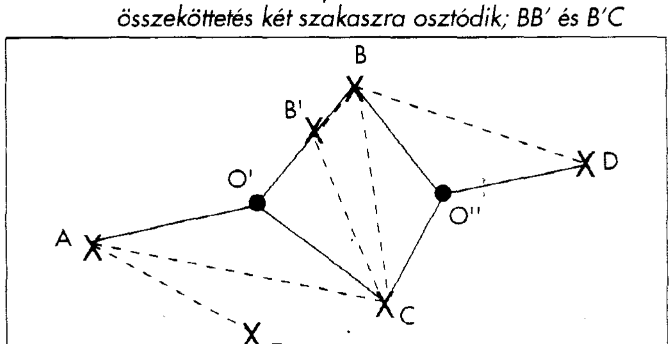
Nincs Y konfiguráció a végpontokon

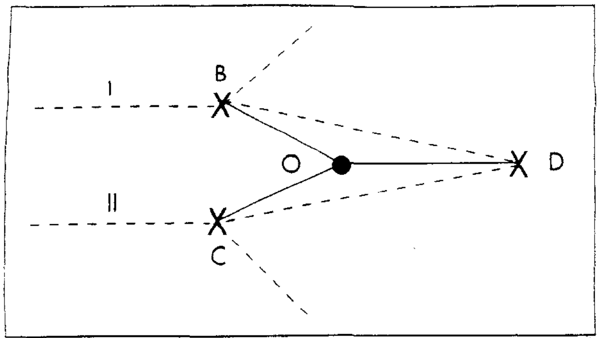
A fentiekben már láttuk, hogy a csomópont-függetlenség követelménye biztosítja a vezeték megosztás elkerülését a

7/a. ábra Két szomszédos Y konfiguráció, amely több összeköttetést tesz lehetővé B és C között

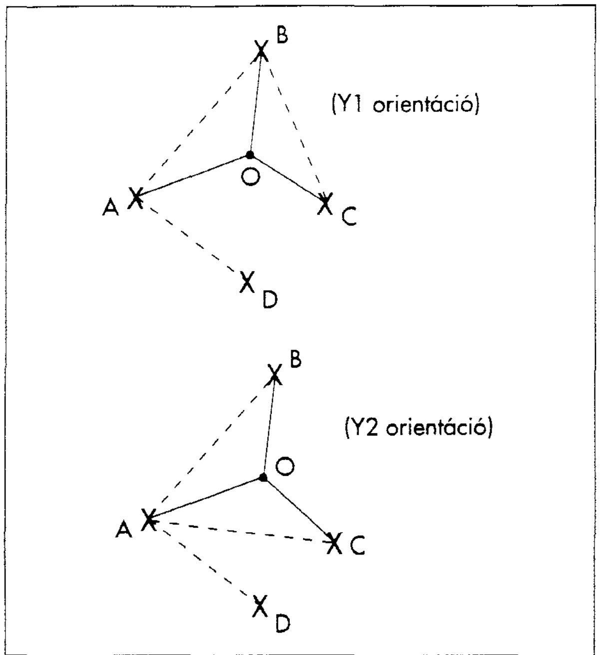


7/b. ábra A B′ vak csomópont bevezetése azt eredményezi, hogy eltűnnek az alternatív összeköttetések, az O′ csatlakozási ponton keresztül vezető BC összeköttetés két szakaszra osztódik; BB′ és B′C

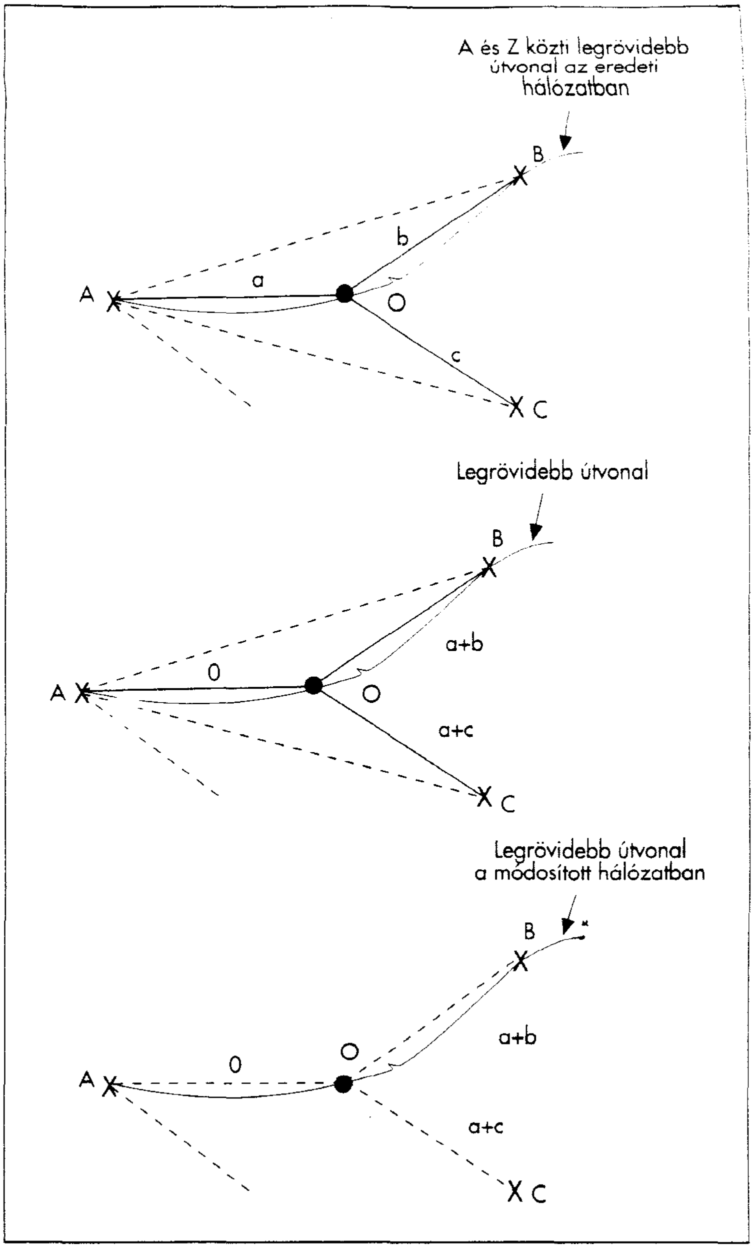




8. ábra Az I és II útvonal Y konfigurációban találkozik



9. ábra Az Y konfiguráció két, Y1 és Y2 orientációja az A végpontban



10. ábra Az Y konfiguráció végpontján lévő, A csomópontból kiinduló és AO vezetéken keresztül haladó legrövidebb útvonal és az ezt követő hálózat-módosítások

csomópontokon kívül. Amennyiben nem jelennek meg Y konfigurációk a végpontokon, az SPNP algoritmus összeköttetés-szintű alkalmazás garantálni fogja az útvonalak közül a legrövidebb fizikailag független (csomópont-független és vezeték-független is) útvonalat.

Y-konfiguráció a végpontokon

Amikor Y konfigurációk jelennek meg a végpontokon, akkor azok két orientációval jelennek meg. Ezeket az Y1-el és Y2-vel jelölt orientációkat a *9. ábrán* mutatjuk be.

Y1 orientáció: A csomópont-függetlenség korlátozás összeköttetés szinten biztosítja a vezeték megosztás elkerülését az *A* csomópontban. Például, ha egy útvonal az *A*-ból *AB* útvonalon lép ki, akkor a másik útvonal szükségszerűen az *AD* összeköttetésen keresztül lép ki *A*-ból és az *OB* vezeték sohasem közös az *A* csomópontból kiinduló két útvonal számára. Tehát elégséges az SPNP algoritmus.

Y2 konfiguráció: Ebben az esetben a csomópont-függetlenség nem garantálja a vezetékek szétválasztását. Például az SPNP algoritmussal talált két útvonal áthaladhat az *AB* és *AC* összeköttetéseken, mely esetben *AO* közös lesz. Következésképpen az SPNP algoritmus nem használható. A teljesség kedvéért tekintsük a legáltalánosabb esetet, amikor az Y konfiguráció mindkét (*A* és *Z*) végponton megjelenik. Az *A* és *Z* közti legrövidebb útvonal az alábbi három kategóriák egyikébe sorolható:

1. Nem halad át a végpontokon lévő Y konfigurációkon
2. Az egyik végponton lévő Y konfiguráción halad keresztül
3. Mindkét végponton lévő Y konfiguráción keresztül halad

1. Eset:
Ebben az esetben a második útvonal (melynek csomópontjai függetlenek az elsőtől) nem fog közösen használni egyetlen szakaszt sem az első útvonallal. Következésképpen, amikor az SPNP algoritmust használjuk, akkor az optimális útvonalat kapjuk meg, amely nemcsak csomópont-független, hanem ág(nyomvonal)-független is.

2. Eset:
Ebben az esetben a második útvonal tartalmazhat az elsővel közös ágat, annak ellenére, hogy csomópont-független az elsőtől (az Y konfiguráció szára az *A* végpontban). A *10. ábrán* láthatjuk az *A* és *Z* közti legrövidebb útvonalat, amely az *AB* összeköttetésen halad keresztül. Nyilvánvaló, hogy az SPNP algoritmussal meghatározott második útvonal áthaladhat az *AC* szakaszon és ily módon közösen használja az *AO* vezetéket a folya-

mat során. Azt állítjuk, hogy ezt a helyzetet ki lehet küszöbölni ha a hálózatot következők szerinti módosítjuk:

a) Helyettesítsük a végponton lévő Y konfigurációt (amelyiken a legrövidebb útvonal keresztül halad) egy olyan új Y konfigurációval, amelyben az Y konfiguráció szára 0 hosszúságú és mindegyik elágazás hossza megegyezik az eredeti összeköttetések hosszával (10. ábra).

b) Helyettesítsük az Y konfiguráció ág-csomópontját (amelyen a legrövidebb út keresztül halad) egy olyan hálózati csomóponttal, hogy az Y konfiguráció ágai a hálózat nyomvonalai legyenek. (10. ábra).

A b) hálózat-módosítás biztosítja, hogy ne jelenjenek meg közös szakaszok (az eredeti hálózatban), mivel az egyetlen ágat, amely a két útvonal számára közös lehet, összeköttetéssé transzformáltuk. Továbbá, mivel ezen összeköttetés hossza 0 az a) módosítás értelmében, az *A* és *Z* közti legrövidebb út változatlan marad a módosított hálózatban. Ez a konvergencia kritikus feltétele, és biztosítja, hogy a független útvonalak optimálisak legyenek, amikor az SPNP algoritmust alkalmazzuk [9].

3. Eset:

A harmadik esetben, amikor a legrövidebb útvonal az Y konfiguráció mindkét, *A* és *Z*, végpontján keresztülhalad, a fenti transzformációt mind az *A* mind a *Z* végpontok mindegyikére végre kell hajtani.

Az Y konfiguráció algoritmusai

Az SPNP két korábbi változatának algoritmusait az alábbi algoritmusba lehet összefoglalni:

1.sz. Algoritmus:

Olyan hálózatban, amely Y konfigurációkat tartalmaz (közvetlen összeköttetések nincsenek), egy adott A és Z csomópont közti fizikailag független útvonal nyomvonalat az alábbi lépésekkel lehet megtalálni:

1. Keressük meg az *A*-tól *Z*-ig vezető legrövidebb útvonalat!
2. Vizsgáljuk meg az így meghatározott legrövidebb útvonal végponti nyomvonalait! Amennyiben valamely végpont az Y konfiguráció szárát képezi, hajtsuk végre a következő transzformációt: Helyettesítsük az Y konfiguráció megfelelő szakaszát egy csomóponttal és változtassuk az Y konfiguráció szárának hosszát zérusra, miközben az egyes ágak hosszát az összeköttetések hosszára növeljük (10 ábra).
3. Módosítsuk a hálózatot az SPNP algoritmus rutinja szerint!
4. Hajtsuk végre ismét a legrövidebb útvonal algoritmust!
5. Töröljük ki az átfedő részeket és közösítsük a szétválasztott csomópontokat, ugyanúgy, mint az SPNP algoritmusban, amivel megkapjuk a csomópont-független útvonalak legrövidebb nyomvonalát. Az így nyert nyomvonal egyben nyomvonal-független is.
6. Transzformáljuk vissza az eredeti hálózatot oly módon, hogy kicseréljük a 2. lépésben bevitt vezeték-csomópontokat, és visszaállítjuk az egyes nyomvonal-hosszakat az eredeti Y konfigurációban szereplő értékekre. Az 5. lépésben nyert elrendezés átalakul az eredeti hálózat optimális változatává.

Hallgatólagos feltételezésünk szerint az összeköttetések az összetevő vezetékek teljes hosszával súlyozottak. Ennek következtében az optimalitás a vezeték hosszra vonatkozik. Fontos itt megjegyezni, hogy ezen algoritmusok elég általánosak ahhoz, hogy más fizikai mennyiséghez – mint például az átvitel dollárban számított ára –, is hozzá lehet rendelni a súlyozást, mely esetben az optimalizálás az adott fizikai mennyiségre például árra vonatkozik.

Az általános villa konfiguráció algoritmusai

Az 1. és 2.sz. Algoritmusokat, amelyeket speciálisan olyan hálózatokra fejlesztettek ki, amelyekben villa konfiguráció fordul elő, olyan hálózatokban is lehet alkalmazni, amelyek kettőnél több ágban tartalmaznak villa konfigurációt (2/b ábra).

Általános algoritmus

A 4. ábra szerinti általános hálózatra alkalmazható algoritmust nyerünk, ha a közvetlen összeköttetést és a villa konfigurációt leíró fejezetekben leírt eredményeket kombináljuk. Mivel a közvetlen összeköttetés fejezetben kapott eredmények a nyomvonal-hosszakra optimalizálnak, az általános algoritmus is erre optimalizál.

2.sz. Algoritmus:

Amikor egy hálózat közvetlen összeköttetéseket és villa konfigurációkat tartalmaz, adott végpont nyomvonal közti fizikailag független útvonal nyomvonalat az alábbi lépésekkel lehet megtalálni:

1. Vegyük ki a hálózatból a közvetlen összeköttetéseket!
2. Hajtsuk végre az 1.sz. Algoritmus lépéseit!
3. Illesszük össze a két útvonal csomópontjait, hogy közvetlen összeköttetéseket alakítsunk ki, ha lehetséges! Ezen összeköttetéseknek azon közvetlen összeköttetések halmazába kell tartozni, amelyeket az 1. lépésben kivettünk a hálózatból.

A 3 lépés haszna abban áll, hogy lecsökkenti az összeköttetések számát az algoritmusban meghatározott két fizikailag független útvonalban. A vezetékek teljes hossza változatlan marad.

ÖSSZEFOGLALÁS

Cikkünkben azokat a hálózatokat vizsgáltuk, amelyeket összeköttetésekkel és nyomvonalnak nevezett fizikai megvalósításokkal lehet leírni. Két különböző összeköttetés megosztozhat ugyanazon nyomvonalon. Megadtunk egy algoritmust az adott csomópont-nyomvonalhoz tartozó legrövidebb fizikailag független nyomvonal meghatározására. A kidolgozott algoritmus-hálózat módosításokat igényel. A független nyomvonalak előnyösek az üzleti szolgáltatások céljára, ezen túlmenően – mivel igen nagy sebességűek – számítástechnikai célra is felhasználhatók real-time kapcsolt környezetben. Továbbá, fel lehet azokat használni a robusztus távközlési hálózatok tervezésében, amely azon koncepción alapul, hogy a hálózat minden csomópont-nyomvonala között a forgalmi folyamot két független útvonalra tereljük.

Ramesh Bhandari

Irodalom

[1] J.W. Suurballe: "Disjoint Paths in a Network", Networks, 4. (1974.)

[2] J. W. Suurballe: "A Quick Method for Finding Shortest Pairs of Disjoint Paths", Networks, 14. (1984.)

[3] R. Bandhari: "Simpler Link/Node-disjoint Shortest Pair Algorithms", (publikáció alatt)

[4] L. R. Ford and D. R. Fulkerson: "Flows in Networks", Princeton University Press (1962.)

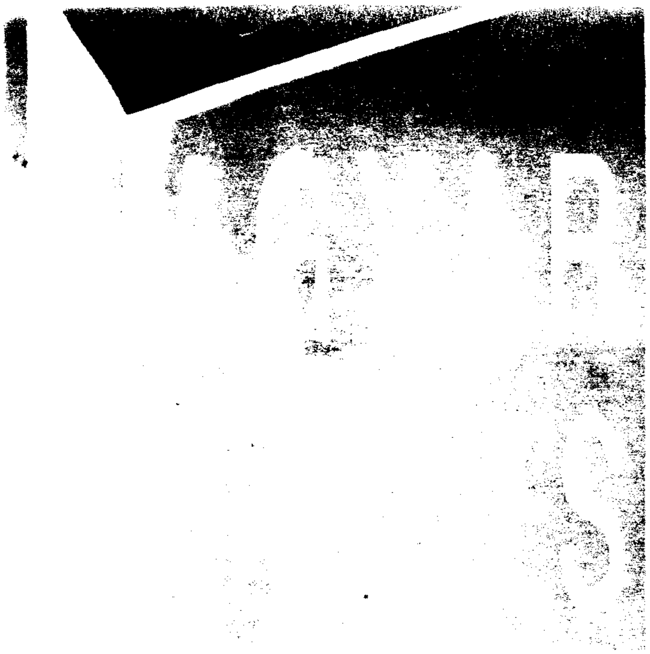
[5] E.W. Dijkstra: "A Note on Two Problems in Connexion with Graphs", Numer. Math. 1. (1959.)

[6] M. Gondran and M. Minoux: "Graphs and Algorithms", John Wiley (1984.)

[7] *A harmad- vagy annál kisebb fokú csomópontok felosztása kevésbé redundáns, amit az [1,2]-ben nem ismertek fel. Egyértelmű, hogy ha ezen csomópontok nem kerülnek felosztásra, az jelentősen csökkenti az olyan ritka hálózatok módosításainak mennyiségét, mint amilyenek az üvegszálas távközlési hálózatok.*

[8] *Habár az Y konfiguráció három lehetséges orientációját mutatjuk be a 8. ábrán, ezek közül alapjában csak kettő különbözik és a harmadik konfiguráció alapjában ugyanaz.*

[9] *A formális bizonyításokat a kézirat terjedelmének korlátozása miatt nem közöltük.*



amikor az elméletek gyakorlattá válnak

95/7

A MAGYAR TÁVKÖZLÉSI RT. FOLYÓIRATA VI. ÉVFOLYAM, 7. SZÁM 1995. JÚLIUS