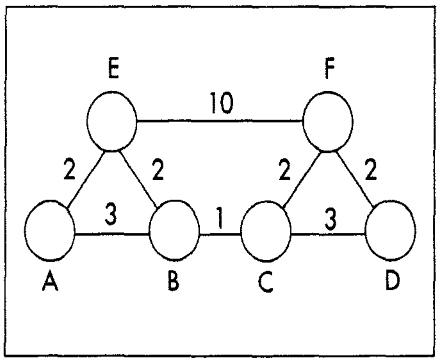
Fizikai kerülőutak kontra hálózati költségminimalizálás

válnak. A cikkünkben vizsgált hálózatoknál a kerülőút adott csomópont-pár között létesített két szétválasztott útra vonatkozik.

Az üzleti felhasználó számára, aki nem tud elfogadni semmilyen késleltetést adatainak átvitelében, a kerülőút biztosítja a szolgálat folyamatosságát egy összeköttetés vagy csomópont hibája esetén. A kerülőút azonban megnöveli a költségeket, mivel számára két fizikailag szétválasztott áramkört kell biztosítani.

Ha egy pillanatra feltételezzük, hogy a szolgálat létesítésének költsége arányos az út hosszával, az előfizetői költségek valószínűleg közel duplájukra nőnek. Ez abból következik, mint az egyedüli (legrövidebb) út kétszerese a hálózat két csomópontja között. Mi történik, ha az ügyfél nem hajlandó ilyen növelt költséget fizetni, ugyanakkor megkívánja a kerülőút lehetőségét, azaz tudunk-e neki két olyan utat létesíteni, sokkal olcsóbban, mint a teljesen független útvonal, amelyek fizikailag közel szétválasztottak.

A bemutatott algoritmus megengedi a megtakarítások értékelését arra az esetre, amikor a két kívánt útnál parciális átfedést engedünk meg. Világos, hogy a két út költsége maximális, amikor teljes (100 százalékos) a függetlenség és minimális abban a szélső esetben, amikor a két áramkör ugyanazon (legrövidebb) úton jön létre, ami a zérus tartalékolás esete. Ezért, amikor a kerülőút biztonsága alacsonyabb, mint 100%, a költség a fenti két szélsőséges eset között helyezkedik el. Valóságban olyan helyzetek állnak elő, amikor a hálózat jellemzőitől függően a költségcsökkentés jelentős lehet a kerülőút függetlenségének csekély csökkenése esetén. Ezt mutatja az *1. ábra*, amely olyan hálózatot mutat be, amelyben fizikailag szétválasztott utakra van szükség a különböző csomópontpárok között.



1. ábra
Csomópontok és kapcsolatok hálózata (a számok a szakaszok költségeit mutatják)

A és *D* csomópontok esetére *ABCD* a legrövidebb (legolcsóbb) út 3+1+3=7 egységnyi költséggel. Az *A* és *D* csomópontok között a két fizikailag független ... biztonság mérséklését, akkor a két áramkör létrehozása – az egyik *ABCD* úton, a másik az *AEBCFD* úton –, a rövid közös *BC* szakaszt eredményezi egy 7+9=16 egységnyi közös teljes költséggel, ami öt egységgel kisebb mint a teljes függetlenség 21 egységnyi költsége. Hasonlóképpen a *B* és *F* között a szétválasztott út legrövidebb párjának költsége *(BCF, BEF)* 3+12=15 egység, míg a két út *(BCF, BCDF)*, amelyre nézve a rövid *BC* szakasz közös, a költség csupán 3+6=9 egység. Így a két utóbbi útpáron áthaladó két áramkör biztosítása (15-9)/15×100% = 40%-kal olcsóbb.

A hálózati költségminimalizálás adott csomópont-pár közötti csomóponttól szétválasztott legrövidebb útpárra vonatkozó algoritmus általánosítására alapozódik. Megadva a hálózati gráfot, G(N,L)-t, ahol N és L jelölik a hálózat csomópontjainak és szakaszainak készletét, a csomópont szétválasztás automatikusan hozza magával a szakasz-szétválasztást, és ily módon a fizikai szétválasztást a két út között (kivéve a közös végponti csomópontokat).

A 2. szakaszban áttekintjük a csomóponti szétválasztás algoritmusát, amit a szerző korábban készített (Bhandari, 1994.), majd a 3. szakaszban kifejlesztünk egy megvalósítási elvet. A csomóponti szétválasztás algoritmusának megvalósítása ezek után automatikusan vezet a maximálisan szétválasztott utakhoz, amely olyan helyzetekben hasznos, ahol a teljes függetlenség nem érhető el magának a hálózatnak a természete miatt. Például az *EF* szakasz törlése az 1. ábrán, elvezet a teljes szétválasztás lehetetlenségéhez. Az *(AD)*, *(AF)* csomópont-párok számára a *BC* szakasz (nevezik hídnak is) szükségszerűen közös a két útra nézve. Általában, amikor a teljes fizikai szétválasztás nem létezik, a kifejlesztett algoritmus két utat talál, amelyek a legalacsonyabb teljes költségűek, és olyat, amelyben a közös csomópontok és szakaszok száma minimális. Végezetük a 4. szakaszban bemutatjuk, hogy a

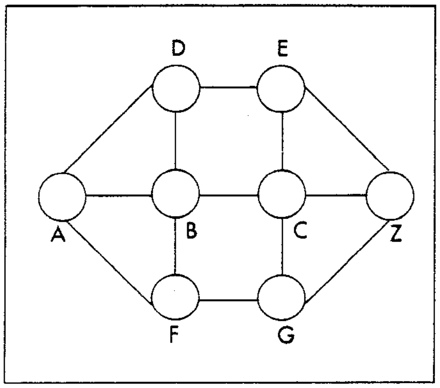
CSOMÓPONTILAG SZÉTVÁLASZTOTT UTAK ALGORITMUSÁNAK LEGRÖVIDEBB PÁRJA

Miután a figyelembe vett hálózat kétirányú, a hálózatot reprezentáló gráf egyirányú, azaz a hálózati gráf minden egyes szakasza egyenértékű két ellenkező irányítású éllel, melynek hossza egyenlő az adott szakasz hosszával. Például az 1. ábrán az *AB* szakasz egyenértékű az *AB* és *BA* élekkel, melyek mindegyikének hossza egyenlő 3-mal, az *AB* szakasz hosszával.

1. algoritmus

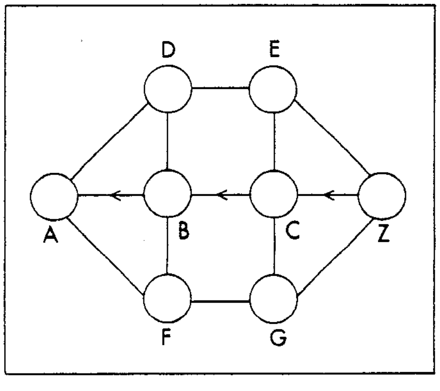
A G(N,L) gráf esetére a csomópontonként szétválasztott utak legrövidebb párja az alábbiak szerint található meg:

A) A figyelembe vett *A* és *Z* adott csomópontpár részére keressük meg a



2/a ábra
Csomópontok és szakaszok hálózata; a legrövidebb útnak ABCZ utat tekintjük, ahol A a forráscsomópont és Z a rendeltetés csomópont

legrövidebb utat a módosított Dijkstra algoritmus használatával. Példaként hivatkozzunk a *2/a. ábrára.* Jelöljük



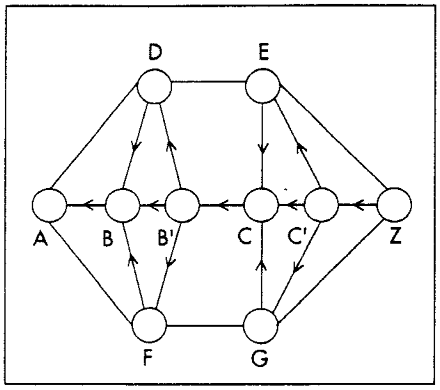
2/b. ábra
Hálózat, amelyben a legrövidebb út szakaszait negatív élekkel helyettesítjük

a talált legrövidebb út hosszát *d(AZ)*-vel, ahol *A* a forráscsomópont és *Z* a rendeltetés csomópont.

B) *A*-tól *Z*-ig töröljük ki az előre irányú íveket minden szakaszra, azaz töröljük ki *AB*-t, *BC*-t és *CZ*-t.

Ez a törlés meghagyja azokat az éleket, amelyek visszirányba mutatnak, azaz *A* forrás csomópont irányába. Tegyük ezeket az ívhosszakat (*BA, CB, ZC* a *2/b. ábrában*) negatívvá.

C) Válasszuk szét a legrövidebb út valamennyi csomópontját (kivéve a végponti csomópontokat) két azonos helyen elhelyezett alcsomóponttá, amelyek zérus hosszúságú éllel vannak összekötve és a forrás csomópont felé irányulnak (a *2/c. ábrában* mind *BB'* és *CC'* zérus hosszúságúak). Helyettesítsük a legrövidebb úton lévő



2/c ábra
Hálózati gráf csomópont szétválasztással módosítva

csomópontok külső szakaszait két éllel, melyeknek hossza ugyanaz, és amint azt a 2/c. ábrán mutatjuk; például a *DB* szakasz a 2/b. ábrán *DB* és *DB'* és hasonlóképpen az *FB, GC* és *EC* szakaszokra.

D) Keressük meg a legrövidebb *A*-tól *Z*-ig terjedő utat a módosított gráfban (lásd a 2/c. ábrát), a módosított Dijkstra algoritmus használatával; jelöljük ennek hosszát *d'(AZ)*-vel.

E) Toljuk át a zérus hosszúságú éleket, azaz egyesítsük az alcsomópontokat szülő csomópontjaikkal. Helyettesítsük a negatív íveket (2/b. ábra) eredeti szakaszaikkal, melyek pozitív hosszúságúak. Távolítsuk el a két út átfedésben lévő szakaszait (az 1. lépésben elsőt és a 4. lépésben a másodikat), hogy megkapjuk a csomópont – szétválasztott utak legrövidebb párját az eredeti gráfban (2/a. ábra).

A B) lépés megengedi a második út – D) lépésben található – összefűzését az A) lépésben található elsővel; például ha a D) lépésben meghatározott

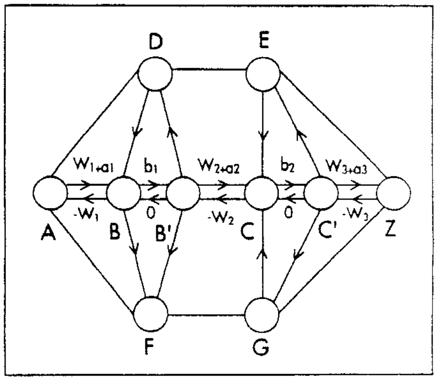
ez lecsökken az eredeti gráfban az *ADECBFGZ* útra (lásd a 2/a. ábrát), és azt mondjuk, hogy ez összefűződik az 1. út *BC* részével, *ABCZ*-vel, amit az A) lépésben találunk; töröljük ki az átfedő részt (vagy az összefűződő részt), *BC*-t, ami ezek után az *(ADECZ, ABFGZ)* párhoz vezet mint legrövidebb szétválasztott úthoz.

Jegyezzük meg, hogy ez az eljárás a C) lépésben leírt csomópont szétválasztás folyamata, ami létrehozza a csomópont függetlenséget a két út között. Ha ezt a lépést elhagyjuk a létrejövő algoritmus megfelel a szakaszszétválasztott utak legrövidebb párjának, ahol a második, most meghatározott utat a módosított gráfban a 2/b. ábra mutatja.

A fentiekben adott 1. algoritmus (Bhandari, 1994.) különbözik az előző változatoktól (Suurbale, 1974.; Suurbale és Tarjan, 1984.). Éles ellentétben Suurbale algoritmusával, a mi algoritmusunk nem igényel kanonikus transzformációt vagy azt, hogy egy általános legrövidebb út algoritmust használjunk mint Ford algoritmusa a fenti D) lépésben. Mi inkább módosítjuk a hagyományos Dijkstra algoritmust, és (Dijkstra, 1959.) kis mértékben módosítva a 2/c. ábra speciális gráfjához jutunk. Ez a módosított Dijkstra algoritmus (Bhandari, 1994.), amit hasznosítunk az 1. algoritmus A) lépésében.

MEGVALÓSÍTÁS ÉS MAXIMÁLIS SZÉTVÁLASZTOTTSÁG ALGORITMUS

A 2/c. ábra mutatja azt a módosított gráfot, amelyben a második út található. Valamennyi él a legrövidebb út mentén, amit az A) lépésben találunk



3. ábra
A 2/c. ábra hálózati gráfja előre

az algoritmusban visszirányba, azaz a r*ɾ* deltetés csomóponttól, *Z*-től a forrás-csomópont, *A* felé mutat. Ez a módosítás az algoritmus B) és C) lépéseinek következménye. A gyakorlatban ahelyett, hogy eltávolítanánk az előreirányú éleket a második lépésben, kényelmesebb ezeket otthagyni, és minden egyes beiktatott zérus hosszúságú élre egy olyan élet előállítani az előre irányban, mint azt a *3. ábra* mutatja.

A 3. ábrán mutatjuk ezeket a hosszúságokat, miszerint $w_i + a_i$, $i = 1, 2, 3$ az *AB, B'C, C'Z* élekre, és egyenlő b_i, $i = 1, 2$ azokra az előre irányú élekre, amelyek kiegészítik a szétválasztott csomóponti éleket zérus hosszúságra; *wi* az eredeti (nem negatív) élhosszúságok, és az a_i (0) és b_i (0) elegendően nagyok ahhoz, hogy ezeket az éleket ne keresztezzük a második út keresése során (az 1. algoritmus D) lépése). Azt is jegyezzük meg, hogy mivel az $a_i > 0$ és $b_i > 0$, nem keletkezik negatív ciklus egy adott előre irányú él és a neki megfelelő visszirányú él között. Ily módon ezeknek az éleknek a bevezetése nem okoz változást az 1. algoritmusban. Amikor azonban teljes fizikai szétválasztás hiányzik, néhány ezekből az előre élekből szükségszerűen keresztezné egymást. Az a_i és b_i paramétereknek megfelelően kiosztott értékek ekkor a maximálisan szétválasztott utak egy algoritmusához vezetnek, ami meghatározza a közös csomópontok és linkek számát, amikor a teljes szétválasztás hiányzik.

2. algoritmus

A csomópontilag szétválasztott utak legrövidebb párjára az algoritmus (1. algoritmus) újra fogalmazható az alábbiak szerint:

A) A figyelembe vett *AZ* adott csomópontpár részére keressük meg a legrövidebb utat *A*-tól *Z*-ig; jelöljük a talált legrövidebb út hosszát *d(AZ)*-vel.

B) Módosítsuk az adott gráfot a csomópontok szétválasztásával az 1. algoritmus B) és C) lépései szerint. Adjuk meg az előre irányú éleket (*A*-tól *Z*-ig terjedő irányú). Illusztrációként lássuk a 3. ábrát.

C) Osszunk ki egy $b_i = b_0$ hosszúságot minden egyes előre él részére, amelyek a szétválasztott csomópontpárokat csatlakoztatják; osszunk ki egy 0 hosszúságot minden egyes megfelelő pontpárnak a visszirány-

D) Osszunk ki egy $a_0 + w_i$ hosszúságot az *i* előre éleknek, melyeknek eredeti hosszúsága w_i volt. Megfelelőjének a visszirányban osszuk ki $a - w_i$ távolságot.

E) Futtassuk a módosított Dijkstra algoritmust a fent módosított gráfon; jelöljük a talált legrövidebb út hosszát *d'(AZ)*-vel.

F) Transzformáljuk az eredeti gráfra. Távolítsunk el bármely átfedő szakaszt a két útból, amelyek keresztezik a legrövidebb út szakaszait az ellenkező irányban.

A kívánt optimális útpárt, amely tartalmazza a két pár úthosszúságát = *d(AZ)* + *d'(AZ)* képletet eredményezi.

1. Következmény: A 2. algoritmus talál egy olyan útpárt, amely maximálisan szétválasztott, azaz egy olyan útpárt, amelyben a közös csomópontok és linkek száma minimális és amelyeknek összege, $d(AZ) + d_0$, minimum; d_0 a módosított gráfban talált második út valódi hossza.

(A számítást lásd: Ramesh Bhandari: „Physical diversity versus cost algorithm for networks." The proceedings of the communication networks modeling and simulation conference (the society for computer simulation), San Diego, California, Jan. 14-17. 1996.; Vol.28. No.1.)

FIZIKAI SZÉTVÁLASZTÁS KÖLTSÉGE

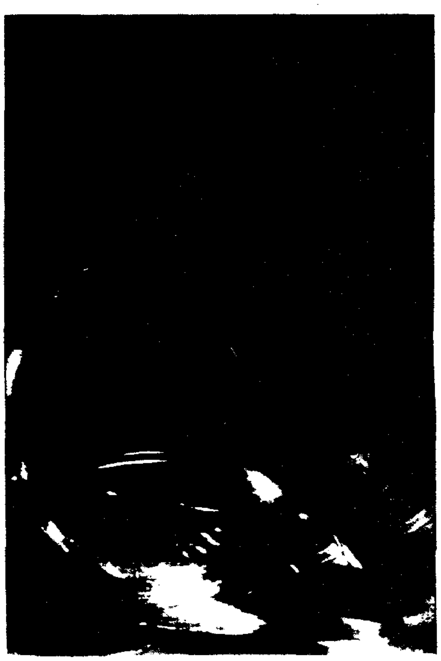
A megelőző szakaszokban leírtuk a fizikai szétválasztás lehetőségeit. A kulcsfontosságú paramétere a_i és b_i voltak, amelyeket az előre élekre osztottak ki *(3. ábra)*. Ezeknek az értékeknek a nagysága biztosította a fizikai szétválasztottságot. Ebben a szakaszban ahelyett, hogy értéküket egy nagy számban, a_0 és b_0 határoztuk volna meg, hagytuk őket szabadon változni, azaz hagytuk $0 \leq a_i \leq a_0$; $0 \leq b_i \leq b_0$ értéken. Ez megengedi a csomópont közösséget, valamint a szakaszközösséget még akkor is, amikor fizikai szétválasztott utak léteznek. Eredményként ezek az utak rövidebbek (vagy kevésbé költségesek), mint a teljes fizikai szétválasztás útjai.

2. Következmény: A b_i-t extra költségnek (büntetésnek) nevezve, az *i* közös csomópontra, és a_i-t extra költségnek (büntetés) nevezve, az *i* szakasszal való átfedésre, és feltételezve, hogy $0 \leq b_i \leq b_0$ és $0 \leq a_i \leq a_0$, a 2. algoritmus útpárokat talál minimális teljes költséggel; az utak rendelkezhetnek közös csomópontokkal és függően; a teljes költség magában foglalja a_i és b_i értékeit, amelyek megfelelnek a közös szakaszoknak és csomópontoknak; a valóságos költség (vagy hosszúság) másik út esetére $d_0 = d'(AZ) - \Sigma a_i - \Sigma b_i$, ahol a szumma a közös szakaszokra és csomópontokra érvényes.

A kidolgozott eljárás megadja a csomópontok szétválasztásának legrövidebb párjait. Ez a csomóponti szétválasztási algoritmus ezután módosul a meglehetősen nagy hosszúságú előre élek kijelölésével. Ezek kijelölése nagy rögzített hosszakat oszt ki az előre irányú élek részére, ami egy maximálisan szétválasztott optimális utat eredményez, s ez megengedi a közös csomópontok és szakaszok számának meghatározását, amikor a teljes fizikai szétválasztás nem lehetséges a hálózat struktúrája következtében. Ez az algoritmus nagyon hasznos lehet a költségmegtakarítás megbecslésére olyan ügyfelek esetében, akik nem tudják fölkínálni a teljesen független fizikai kerülőút költségét.

A minimális költségű út számításait először a szerző 1991-ben közölte „Legkisebb költségű algoritmus egy útpár esetére" („A Least Cost Algorithm for a Pair of Paths"; February 21, 1991, nem publikált) címen, melyet azóta kibővített úgy, hogy a fizikai szétválasztási problémák bonyolultabb eseteivel is foglalkozik a valós élet hálózataiban (Bhandari, 1994.).

Sajtó alá rendezte:
Horváth Imre



MAGYAR TÁVKÖZLÉS

távközlés és politika: egymásra utalva

96/7

A MAGYAR TÁVKÖZLÉSI RT. FOLYÓIRATA

VII. ÉVFOLYAM, 7. SZÁM 1996. JÚLIUS