Optimális túlélőhálózatok

ALGORITMUSOK DISZJUNKT ÚTVONALRA

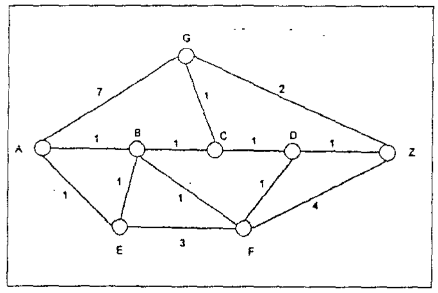
Egy hálózat megbízhatósága növelhető a fizikai diverzitivel, amikor a hálózatban adott csomópont-pár között a forgalmat két vagy több fizikailag elkülönített útvonalon vezetjük. Másik megoldás, ha elegendő tartalék kapacitást rendelünk az egyes útvonalakhoz, ekkor valamely csomópont vagy fizikai kapcsolat meghibásodásakor a forgalmat azonnal át lehet irányítani előre meghatározott útvonalakra. Cikkünk egy csomópontokból és élekből álló gráfban *K* diszjunkt útvonalra ad optimális algoritmusokat (K≥2).

Mivel a hálózatokban egyre nagyobb mértékben alkalmaznak fényvezetőszálat, a hálózat használhatósága ma fontosabb, mint valaha. Egy adott hálózat megbízhatóságát növelhetjük például fizikai diverziti útján, ami azt jelenti, hogy a hálózatban adott csomópontpár közötti forgalmat két vagy több fizikailag elkülönített útvonalon vezetjük, így ha egy csomópont vagy fizikai kapcsolat az egyik diszjunkt útvonalon meghibásodik, nem okoz teljes forgalomvesztést. Alternatív lehetőségként kínálkozik az a megoldás, amelynek során elegendően nagy tartalék kapacitást osztunk ki az egyes útvonalakhoz, és így egy csomópont vagy fizikai kapcsolat meghibásodása esetén a különben elvesző forgalmat át lehet irányítani előre meghatározott diszjunkt útvonalakra.

A hálózatokat (gráfokat) csomópontokkal és élekkel reprezentáljuk. Eltérő utalás hiányában feltételezzük, hogy több él nincs jelen. Az *1. ábrán* példát mutatunk be egy 8 csomópontból és 13 élből álló kétirányú hálózatra; minden él ekvivalens két ellentétes irányú ívvel és mindegyik hossz megegyezik az élhosszal. A hossznak itt általános jelentése van, amennyiben reprezentálhatja egy él fizikai hosszát a hálózatban, de jelentheti egy él használatának költségét is. Amikor különféle útvonalakat keresünk adott csúcspárok között, célszerű olyan útvonalakat kiválasztani, amelyek összege minimális. Ha például egy él hossza egy fizikai kapcsolatot alkotó fényvezetőszál hossza, akkor az optimális útvonal minimális mennyiségű fényvezetőszálat vesz igénybe azt adott csomópont-párok között.

DISZJUNKT ÚTVONALI ALGORITMUSOK

Az egyszerű módszert megvalósító algoritmusok megkeresik az első legrövidebb útvonalat az adott csúcspontpárok között, majd a második legrövidebb útvonalat, amely elkülönül az elsőtől, majd a harmadik legrövidebb útvonalat, amely elkülönül az első kettőtől és így tovább, a kívánt diszjunkt útvonalak számától függően. Egy diszjunkt útvonal-párra koncentrálva: amennyiben diszjunkt kapcsolatra van szükség, a gráfból az első legrövidebb útvonal összes kapcsolatát törölni kell, majd a legrövidebb útvonal algoritmusát ezen a redukált gráfon kell lefuttatni. Hasonlóképpen: ha a második útvonal csúcspont-diszjunkt, akkor az első legrövidebb útvonal csúcspontjaihoz tartozó kapcsolatokat (a végponti csúcsok kivételével) el kell távolítani. Ezeket az algoritmusokat az 1. ábrára történő hivatkozással illusztráljuk. Tegyük fel, hogy egy diszjunkt útvonal-párra van szükség *A* és *Z* csúcspontok között. Ekkor *A* és *Z* között a legrövidebb útvonal a 4 hosszúságú *ABCDZ*. A második legrövidebb útvonal, amely él szempontjából diszjunkt az elsővel, a 7 hosszúságú *AEFBZ*, amely az él-diszjunkt útvonal párok tekintetében összesen 11 hosszúságot képvisel. Hasonlóképpen, az 1. ábrán a legrövidebb csúcs-diszjunkt útvonal az *ABCDZ* útvonalhoz képest az *AEFZ=8*, ami a csúcs-diszjunkt útvonal-párokra nézve 12 teljes hosszat jelent. A diszjunkt útvonalak keresésére ismertetett egyszerű megoldásnak azonban van néhány hiányossága:



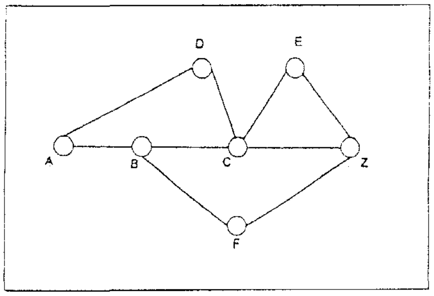
1. ábra *Kétirányú hálózat; a számok az élhosszt jelentik*

❶ *Szuboptimalitás.*

Látnunk kell, hogy az 1. ábrán a legrövidebb él-diszjunkt útvonalpár voltaképpen *(ABCGZ, AEBFDZ)*, melyek teljes hossza 10. Hasonlóképpen, a legrövidebb csúcs-diszjunkt útvonalpár *(ABCGZ, AEFDZ)*, melyek teljes hossza 11. Amennyiben az egyszerű algoritmust alkalmazzuk az 1. ábrára, nem kapjuk meg az optimális diszjunkt útvonalpárokat *A* és *Z* között. Az ilyen típusú algoritmusok általában vagy megadják az optimális diszjunkt útvonal-készletet, vagy nem.

❷ *Hamis riasztások nem létező útvonalak miatt olyankor, amikor ilyen útvonalak léteznek.*

Ezt a hiányosságot a *2. ábrán* illusztráljuk. Tegyük fel, hogy *ABCZ* a legrövidebb útvonal adott *A* és *Z* csúcspárok között; az egyszerű algoritmus ilyenkor nem talál csúcs-diszjunkt párt, jóllehet két ilyen pár is létezik: *(ABFZ, ADCZ)* és *(ABFZ, ADCEZ)*. Nyilvánvalóan az egy-



2. ábra *Az egyszerű megoldású algoritmus nem képez csúcsdiszjunkt útvonalpárokat A és Z között*

szerű szemléletű algoritmus bármely gyakorlati megvalósítása azzal a kockázattal jár, hogy esetleg hamis riasztást generál diszjunkt útvonalak hiánya miatt olyankor is, amikor valójában működnek útvonalak. Másként fogalmazva: kerülni kell, hogy a diverz útvonalakat igénylő ügyfél tévedésből olyan információt kapjon, hogy ilyen kapcsolat számára nem létesíthető.

LEGRÖVIDEBB K (≥2) ÚTVONALI ALGORITMUS

A legrövidebb K(≥2) diszjunkt útvonalakra már korábban is ismertek voltak algoritmusok [1, 2]. Ezek az algoritmusok azonban a különleges kanonikus transzformáció használatát hangsúlyozzák, s ezzel szükségtelenül bonyolult algoritmusok születnek, amelyekkel a gyakorló mérnök nehezen boldogul. Cikkünkben egyszerűbb változatot ismertetünk a diszjunkt algoritmusokra, amely nem igényel speciális hálózati transzformációt, sőt a legrövidebb útvonal megtalálásához csak csekély mértékben kell módosítani a szabványos Dijkstra algoritmust.

Éldiszjunkt legrövidebb útvonal-párok algoritmusa

A gráfban adott csúcspár közötti legrövidebb éldiszjunkt útvonalpár előállítására szolgáló algoritmus egyszerűen megadható a következőképpen [5]:

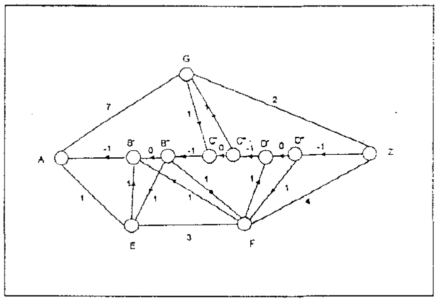
❶ Futtassuk le a kérdéses csúcspárokra (*A* és *Z*) a legrövidebb útvonali algoritmust Illusztrációként lásd az 1. ábrát, amelyen feltételezzük, hogy *A* a forrás csúcs és *Z* a rendeltetési csúcs.

❷ Helyettesítsük a legrövidebb útvonal minden élét a forrás csúcs felé mutató egyetlen ívvel (az 1. ábrán *A* csúcs). Ezzel biztosítjuk, hogy a legrövidebb útvonal (az 1. ábrán a 4 hosszúságú *ABCDZ*) nem reprodukálódik, amikor a legrövidebb útvonali algoritmust lefuttatjuk a módosított gráfon (*3. ábra*).

❸ Tegyük negatívvá a fenti ívek hosszát.

❹ A módosított gráfon futtassuk le a legrövidebb útvonali algoritmust az *A* csúcs felől a *Z* csúcs felé (3. ábra).

❺ A gráfban a negatív íveket helyettesítsük az eredeti (pozitív hosszúságú) élekkel. Távolítsuk el a két útvonal átfedő éleit. Ezzel megkaptuk a kívánt útvonalpárt.



3. ábra *Az 1. ábra módosított változata a legrövidebb éldiszjunkt pár algoritmushoz*

Csúcsdiszjunkt legrövidebb útvonal-párok algoritmusa

A csúcsdiszjunkcióhoz tartozó algoritmusokhoz is kétszer kell lefuttatni a legrövidebb útvonali algoritmust a módosított gráfon, az alábbiakban ismertetett jelentős eltérésekkel, amelyeket az 1. és *4. ábrára* való hivatkozással szemléltetünk. A csúcsdiszjunkcióhoz *A* és *Z* között minden lehetséges útvonalat, amely metszi az *ABCDZ* legrövidebb útvonalat, ki kell zárni a második legrövidebb útvonal lehetőségei közül.

A 3. ábrán például az *AEBFZ* és az *AEBFDCGZ* útvonalak szóba jöhetnének ugyan az éldiszjunkt algoritmusban, a csúcsdiszjunkció szempontjából mégis érvénytelenek. Az ilyen útvonalak kizárása úgy történik, hogy az első legrövidebb útvonal megtalálása után a csúcsokat megosztjuk. A normál csúcsmegosztó eljárás [6] az alábbi algoritmusokat adja a legrövidebb csúcsdiszjunkt útvonalpárokra:

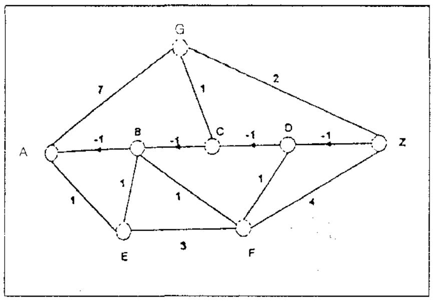
❶ A vizsgált csúcspárra megkeressük a legrövidebb útvonalat az algoritmussal. Példaként nézzük az 1. ábrán látható hálózati gráfot.

❷ A legrövidebb útvonalon helyettesítsünk minden élt a forráscsúcs felé mutató negatív ívvel.

❸ A legrövidebb útvonalon osszunk meg minden csúcsot két közös elhelyezésű alcsúcsra, amelyeket nulla hosszúságú ív köt össze. Irányítsuk ezt az ívet a forráscsúcs felé. Helyettesítsük a legrövidebb útvonalon a csúcsokhoz csatlakozó külső éleket két ellentétes irányú, azonos hosszúságú ívvel és csatlakoztassuk ezeket a két alcsúcshoz a *4. ábrán* látható módon; a külső ívek az egyvonalas alcsúcsokon végződnek és a kétvonalas alcsúcsoktól indulnak.

❹ Futtassuk le a legrövidebb útvonali algoritmust a *4. ábrán* látható módosított gráfon.

❺ Távolítsuk el a nulla hosszúságú íveket; vonjuk össze az alcsúcsokat anyacsúcsaikkal. A legrövidebb útvonalon helyettesítsük az egyszeres íveket az eredeti élekkel. Távolítsuk el a kapott két útvonalon az átfedő éleket. Ezzel megkaptuk a legrövidebb csúcsdiszjunkt útvonalpárt.

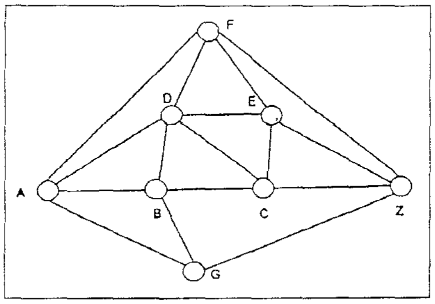


4. ábra *Az 1. ábra módosított változata a legrövidebb csúcsdiszjunkt pár algoritmushoz*

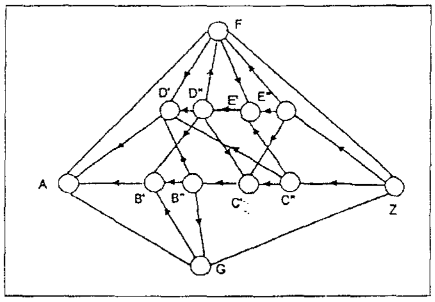
Az 1. ábra szerinti 1-3. lépéssel kezdve megkapjuk a 4. ábra szerinti gráfot. A kapott legrövidebb útvonal az 1+3+1-1+1+2 = 7 hosszúságú *AEFD'C''GZ*, míg a legrövidebb csúcsdiszjunkt útvonalpár teljes hossza 4+7 = 11. Végül az 5. lépéssel azt kapjuk, hogy a legrövidebb csúcsdiszjunkt útvonalpár a 11 hosszúságú (*ABCGZ, AEFDZ*) az 1. ábra szerinti eredeti gráfon.

Csúcsdiszjunkt útvonalak

Tegyük fel, hogy az *5/a. ábrán* a korábban ismertetett algoritmussal kapott legrövidebb csúcsdiszjunkt útvonalpár *A* és *Z* csúcsok között az (*ABCZ, ADEZ*). Ahhoz, hogy triplettet kapjunk, módosítjuk az adott gráfot úgy, hogy az adott legrövidebb útvonalpáron megosztjuk a csúcsokat, és összekapcsoljuk őket a gráf más csúcsaival ugyanolyan szabályok szerint, mint amilyeneket a legrövidebb pár algoritmusának kialakításakor alkalmaztunk. Az *5/b. ábra* a módosított gráfot mutatja, amelyben a legrövidebb útvonali algoritmus *A*-tól *Z*-ig fut. Bármilyen esetleges átfűzés esetén a közös részeket ki kell törölni és az osztott csúcsokat ismét össze kell vonni az anyacsúcsokkal. A végeredmény a csúcsdiszjunkt útvonalak legrövidebb triplettje. Hasonló módon további iterációs lépésekkel háromnál több csúcsdiszjunkt útvonalat kaphatunk minimális teljes hosszúsággal az adott hálózati gráfban, ha léteznek ilyen útvonalak.



5/a. ábra *(ABCZ, ADEZ) a legrövidebb csúcsdiszjunkt útvonalpár*

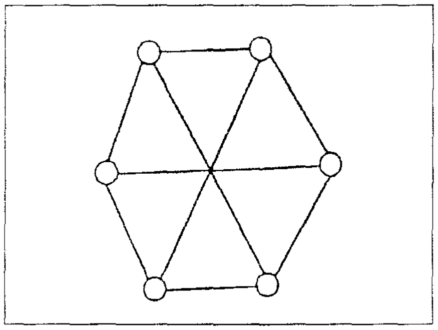


5/b. ábra *Módosított gráf a legrövidebb triplettek meghatározásához*

Túlélő szövevényes hálózatok

A tartalék kapacitás kiosztására számos algoritmus létezik, ezek általában arra épülnek, hogy az érintett forgalmat a meghibásodott kapcsolatot megkerülve átirányítják. A legrövidebb útvonali algoritmust általában alternatív útvonalak felkutatására és tartalék kapacitás megszerzésére használják az egyes kapcsolatokon [7]. Azokban a hálózati rendszerekben, amelyek a forgalmat a meghibásodott szakasz megkerülésével átirányítják, a forgalom helyreállításának ideje jelentős lehet, mert meg kell határozni, hol helyezkedik el a hálózaton belül a hibás kapcsolat vagy csomópont.

A szövevényes hálózatok magas fokú konnektivitásuk miatt magukban rejtik a kapcsolatok kereszteződésének lehetőségét. A *6. ábrán* példaként olyan hálózatot mutatunk, amelyben a fizikai kapcsolatok alkészletei keresztezik egymást. A keresztezési pontok például regenerátor helyek lehetnek. Noha három diszjunkt útvonal van minden csomópont pár között, ez mégsem jó hálózati konstrukció, mert egy központi helyen keletkező sérülés vagy meghibásodás következtében a hálózat kapcsolatainak fele kiesik. Az ilyen hálózati konstrukció tehát egyértelműen kerülendő. A későbbiekben bemutatunk egy módszert a síkba ágyazott optimális hálózati topológiákra. Ez olyan síkbeli hálózati konfigurációt jelent, amely lehetővé teszi adott számú diszjunkt útvonal kialakítását minimális számú, egymást nem keresztező kapcsolattal. A $K = 3$ esetet konkrétan is bemutatjuk.



6. ábra *6 csomópontú (3-csatlakozású) hálózat; a kapcsolatok 50%-a egyetlen pontban metszi egymást*

KAPACITÁSKIOSZTÁS

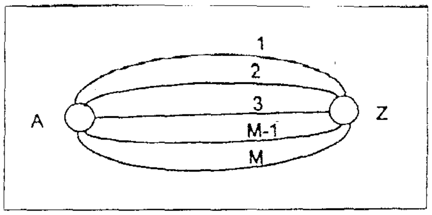
Tegyük fel, hogy egy túlélő szövevényes hálózat tervezésével foglalkozunk, amelynél valamely csomópont vagy kapcsolat meghibásodása esetén követelmény az érintett forgalom 100 %-os helyreállítása. Feltételezzük azt is, hogy a hálózatban az egyes csomópont párok közötti forgalmi igényeket ismerjük. A kapacitáskiosztásra (üzemi és tartalék kapacitás együtt) javasolt modellt az egyes csúcspárok közötti forgalmi igény, valamint az ugyanazon diszjunkt útvonalakra vonatkozó üzemi és tartalék kapacitás kiosztás figyelembevételével mutatjuk be:

a) Amennyiben a *7. ábrán T* jelöli az *A* és *Z* csúcspár közötti forgalmi igényt, akkor az egyes *M* diszjunkt útvonalakon keresztül irányított forgalom *T/M*.

b) Amennyiben a fenti útvonalak valamelyikén egy kapcsolat vagy csomópont meghibásodik, az érintett forgalmat (esetünkben a csomópont párra) = *T/M*) az *A* és *Z* végpontokban elhelyezett digitális rendező rendszerek *(DCS)* átkapcsolják a fennmaradó *M-1* útvonalra. Ez azt jelenti, hogy az *M* útvonal mindegyikén *T/(M(M-1))* tartalék kapacitás van lefoglalva.

Gyors helyreállítási és költségmegszorítási kritériumok M-re

Minél nagyobb számot jelent M, annál kisebb az a forgalommennyiség, amit a csomópont pár között át kell kapcsolni.



7. ábra *A és Z csomópontok, melyek között a T forgalom M diszjunkt útvonalon folyik*

Ebből következik, hogy amennyiben egy DCS rendező átkapcsolási ideje függ az átkapcsolandó forgalom mennyiségétől, akkor a helyreállítási idő csökkentése érdekében célszerű *M* értékét nagyra választani. Nyilvánvalóan *1MK*, ahol *K* a diszjunkt útvonalak maximális száma *A* és *Z* között a 7. ábrán. *M* minimális értéke 2. Ha azonban azt akarjuk, hogy a helyreállítás gyorsan történjen meg hogy az ügyfelek ne vegyék észre a meghibásodást vagy az átállítást, akkor az *M* érték 2-nél nagyobb is lehet.

Tegyük fel, hogy *M* értékét megválasztottuk a fenti gyors helyreállítási kritériumoknak megfelelően. Javítható-e értéke költség szempontból? A kérdés érzékeltetésére feltételezzük, hogy a költség mértéke az útvonalra kiosztott teljes (üzemi + tartalék) kapacitásnak és az útvonal fizikai hosszának a szorzata. A kapacitás határozza meg az útvonalhoz szükséges fényvezető szálak számát, és az útvonal hossza megegyezik az útvonal nyomvonalán lefektetett fényvezető szálak hosszával. A 7. ábra és a már leírtak figyelembevételével egy útvonalon a 100 %-os helyreállításhoz szükséges teljes kapacitást az

$$C_{tot} = T/M + T/(M(M-1)) = \\ = T/(M-1) \qquad (1)$$

adja meg.

A 7. ábrán megadott csomópont-pár közötti forgalomra a teljes forgalom-mérföld értékét a

$$T_{tot}(M) = C_{tot} \sum_{i=1}^{M} l_i = T/(M-1) \sum_{i=1}^{M} l_i, \qquad (2)$$

adja, ahol $T_{tot}(M)$ a teljes forgalommérföld, l_i pedig *i* útvonal hossza. Most tételezzük fel, hogy *M+p* útvonal *(p1)* is létezik. Ekkor a (2) egyenlet a

$$T_{tot}(M+p)/T_{tot}(M) = \\ = (M-1)/(M+p-1) \sum_{i=1}^{M+p} l'_i / \sum_{i=1}^{M} l_i \qquad (3)$$

szerint alakul, ahol l_i' az *i* útvonal hossza az *M+p* diszjunkt útvonalak csoportján belül. A (3) egyenletből

$$T_{tl}(M+p)/T_{tl}(M) \le 1 \qquad (4)$$

lesz, feltéve, hogy

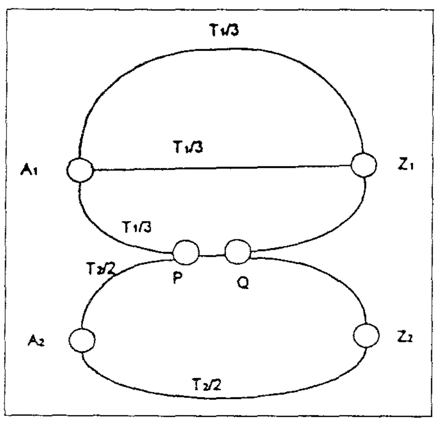
$$\sum_{i=1}^{M+p} l'_i \Big/ \sum_{i=1}^{M} l_i \le (M+p-1)/(M-1) \qquad (5)$$

Amennyiben tehát az (5) egyenlet feltételei teljesülnek, költség szempontból kedvezőbb lenne a forgalmat az *M* diszjunkt útvonalak helyett az *(M+p)* diszjunkt útvonalakon át irányítani. Tehát a következő előnyökkel jár, ha a kapacitást *M* útvonal helyett *M+p* útvonalra osztjuk ki:

❶ Több forgalom küldhető ugyanazon csomópontpár között azonos gyors helyreállítási kritériumok mellett, illetve adott mennyiségű forgalom esetén kevesebb forgalmat kell átkapcsolni egy kapcsolat vagy csomópont meghibásodásakor.

❷ A nagyobb számnak köszönhetően az *M+p* útvonalkészlet nagyobb potenciált kínál, mint az *M* útvonalkészlet a tartalék kapacitás megosztására más csomópont párok útvonalai között.

Ha a fenti folyamatot a hálózat minden csomópont-párján megismételjük, az egyes kapcsolatok üzemi kapacitását a kapcsolattal közös többi csomópont-párnak megfelelő üzemi kapacitások összegzésével kaphatjuk meg *(8. ábra)*. Hasonlóképpen, a tartalék kapacitás kiosztása az egyes kapcsolatokon megosztást is jelent, ez szintén látható a *8. ábrán*. Miközben például a *PQ* kapcsolaton a teljes üzemi kapacitás $T_1/3 + T_2/2$, a tartalék kapacitás maximuma $(T_1/6, T_2/2)$.



8. ábra *Tartalék kapacitás megosztása*

Hálózati topológiák diszjunkt útvonalakra

Felvetődik a kérdés: adva van *N* csomópont, amelyből fel kell építeni a hálózatot. Mi a minimális a topológiája egy K-csatlakozású hálózatnak? Azt a hálózatot nevezzük K-csatlakozásúnak, amely *K* diszjunkt útvonalat tesz lehetővé minden csomópont-pár között. A minimális topológia pedig a lehető legkisebb számú kapcsolatot jelenti, amelyek nem keresztezik egymást. A kapcsolatok keresztezés-mentességét azért szabjuk feltételül, mert a gyakorlatban a legtöbb hálózat földfelszíni hálózat, lényegében síkban fekszenek, és a kereszteződés növelné a hálózat sebezhetőségét, ami pedig éppen a túlélés miatt kerülni kell. Tulajdonképpen egy síkbeli K csatlakozású, minimális számú kapcsolatot tartalmazó gráfot keresünk.

A K-csatlakozás azzal is együtt jár, hogy minden csomópont legalább *K* értékű, vagyis *K* fokú. Ennek fordítottja nem feltétlenül igaz: a *K* érték nem jár szükségképpen K-csatlakozással *(9. ábra)*. Mindazonáltal a minimális számú kapcsolatot tartalmazó K-csatlakozású hálózat építésekor előírjuk, hogy a gráf minden csomópontja *K* fokú. Minden *K* értékű csomópont minimális számú kapcsolathoz vezet

$$E_{min} = NK/2, \qquad (6/a)$$

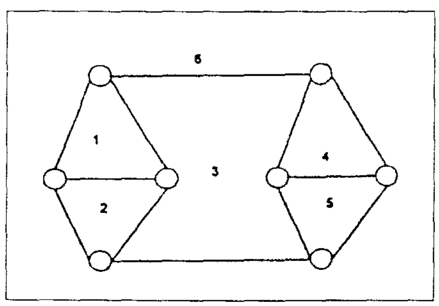
esetén, feltételezve, hogy az *NK* szorzat páros szám; az 1/2-es tényező annak következtében lép be, hogy minden kapcsolat, amely közös valamely csomópont párral, kettővel megnöveli a csomópontok összegének fokát. Ha *N* és *K* egyaránt páratlan, akkor

$$E_{min} = (NK+1)/2. \qquad (6/b)$$

Mivel egy *N* csomópontot tartalmazó gráfban a kapcsolatok teljes száma nem lehet nagyobb, mint *N(N-1)/2* (továbbra sem számolunk többszörös élekkel), K-csatlakozás hálózat esetén az

$$N \ge K+1 \qquad (7)$$

egyenlőtlenséghez jutunk.



9. ábra *Háromértékű, de nem 3-csatlakozású gráf*

Idézzük most fel Euler tételét [4], amely szerint egy síkbeli gráfnak ki kell elégítenie a következő egyenletet:

$$E = N + F - 2 \qquad (8)$$

ahol *E*, *N* és *F* rendre a gráfban lévő kapcsolatok, csomópontok és lapok száma. A lapok területeket jelentenek, melyeket a 9. ábrán számokkal jelölünk. A 6. lap végtelen területnek számít, ezt is figyelembe veszi a 8. egyenlet. Ha most a síkbeli K-csatlakozású gráfoknak a (6/a) és (6/b) egyenlettel megadott minimális számú kapcsolattal kell rendelkezniük, akkor a 8. egyenlet a következők szerint módosul:

$$F = (K-2)N/2 + 2, \qquad (9/a)$$

$$F = (K-2)N/2 - 5/2 \qquad (9/b)$$

ahol *N* és *K* egyaránt páratlan szám.

A továbbiakban *K = 2* és *K = 3* értékre vizsgáljuk meg az optimális topológiákat.

❶ ***K = 2*** A (9/a) egyenletből látjuk, hogy *F = 2*, a megfelelő hálózati topológia egy olyan gyűrű, amely a hálózat összes csomópontján áthalad. Minden csomópont-pár között két diszjunkt útvonal létezik. A kapcsolatok száma a (6/a) egyenlet értelmében *N*, ez egyúttal a csomópontok száma is. A gyűrűn megengedett csomópontok minimális száma a (7) egyenlet szerint *K + 1 = 3* (a gyűrűn 2 csomópont esete */N=2/* egy többélű, 2 csomópontú gráfnak felel meg, amit helyesen tiltanak az egyenleteink).

K = 3 A csomópontok minimális száma a (7) egyenletből *3+1=4*, és ehhez a minimumhoz a kapcsolatok minimális száma a (6/a) egyenletből *3 × 4/2 = 6*. Ugyanígy *F=N/2+2=4*. Ezek a feltételek *N=4* esetén 4 csomópontú hálózattal teljesülnek. Minden csomópont párhoz három diszjunkt útvonal tartozik.

Ramesh Bhandari

Irodalom

[1] J. W. Suurballe, "Disjoint Paths in a Network", Networks, 4 (1974) 125-145.

[2] J. W. Suurballe and R. E. Tarjan, "A Quick Method for Finding Shortest Pairs of Disjoint Paths", Networks, 14 (1984) 325-336.

[3] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graps", Numer.Math. 1 (1959) 269-271.

[4] M. Gondran and M. Minoux, Graphs and Algorithms, John Wiley (1984).

[5] Proofs of the algorithsm are not been given here due to brevity constraints.

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

[7] T. Wu, Fiber Network Service Survivability, Artech House (1992).

[8] B. Grunbaum, Convex Polytopes, Interscience Publishers (1967).