Az Európai Unió légihálózatának elemzése gráfelméleti módszerekkel

Oláh Kitti*

*Óbudai Egyetem, Mechatronikai és Járműtechnikai Intézet, 1081 Budapest, Népszínház utca 8.

(e-mail: olah.kitti@bgk.uni-obuda.hu)

Absztrakt: Manapság észre sem vesszük, hogy a mindennapjainkat körül veszik a gráfelméleti törvényszerűségek. A gráf, mint modellezési forma jelen van a természettudományokban, számítógéptudományban, gazdaság-és társadalomtudományban és a közlekedésben is. A légiközlekedés kritikus infrastruktúrák közé sorolható, ezért különösen figyelni kell a hibatűrő képességére. Az Európai Unió (EU) tagállamaihoz tartozó nemzetközi reptérhálózat, hibatűrő képességét vizsgálom, amihez a többszörös összefüggőséget, mint gráfelméleti elemet fogom használni.

1. BEVEZETÉS

A hálózatokat különböző matematikai módszerekkel tudjuk elemezni, aminek legalapvetőbb eszköze a gráfelméletet. A gráf, pontok és élek halmazából tevődik össze, amelynél az élek felelnek egyes csomópontok között lévő kapcsolat megjelenítéséért. Az általam kiválasztott légihálózatnak a csomópontjai lesznek az országok és az őket összekötő járatok pedig az élek. A többszörös összefüggőség megmutatja, hogy mennyire ellenálló támadásokkal szemben egy gráf, azaz hogy maximum mennyi élt lehet eltávolítani ahhoz, hogy a hálózat továbbra is összefüggű maradjon. A cikkben kitérek arra is, hogy milyen hatással bír egy-egy él eltávolítása. Végül a számításhoz szükséges gráfalgoritmusokat programozói felületen fogom megvalósítani.

2. GRÁFELMÉLETI ALAPFOGALMAK

A cikk teljes megértéséhez szükséges néhány fogalom ismertetése.

Egy gráfot két halmaz alkot: G = (V, E), ahol a V halmazba tartoznak a gráf csúcsai/pontjai és az E halmazba pedig a gráf élei. A csúcsok kételemű részhalmazai tartalmazzák az éleket.

Irányított egy gráf, ha élei rendezett párok, amiket nyilakkal jelölünk. Hurokélnek nevezzük azokat az éleket, amiknek a kezdő- és végpontjaik ugyanazok.

Az összetett gráfok többszörös éleket is tartalmaznak, amik kettő vagy több eltérő, nem hurokélnek a végpontjai egyformák. Fokszám a csúcsokba vezető élek összességét jelenti. Az út az éleknek azon sorozata, mely egy csúcsot legfeljebb csak egyszer érint.

Éldiszjunkt utaknak nevezzük az összes olyan utat, amelyeknek nincsenek közös élei.

Pontdiszjunkt utaknak nevezzük az összes olyan utat, aminek a kezdő-és végpontjai kivételével nincsenek közös pontjaik.

Egy G gráf összefüggő, ha bármely két csúcsához tartozik csúcsot összekötő út a gráfban, ezen utak a G gráf részgráfjai. A H (V', E') gráf a G (V, E) részgráfja, ha $V' \subseteq V$ és $E' \subseteq E$ illetve, egy csúcs vagy él pontosan akkor illeszkedik egymásra H gráfban, ha G-ben.

Artikulációs pontok halmazába rendeljük azokat a pontokat egy összefüggő gráfban, amiket elhagyva a gráf elveszti összefüggőségét.

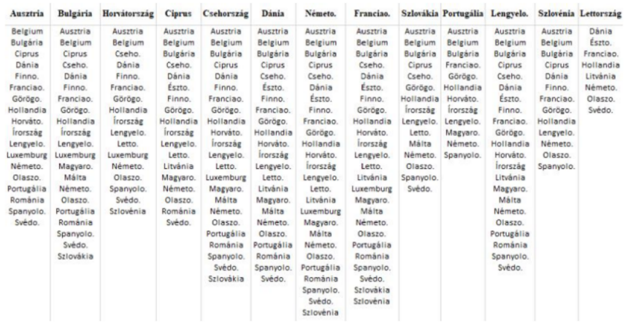
Elvágó élek halmaza tartalmazza az összes olyan élt, amely eltávolításával a gráf elveszti összefüggőségét.

A többszörös összefüggőségnek két típusát ismerjük. Egy gráf lehet k-szorosan pontösszefüggő, ha legalább k+1 csúcsa van és bárhonnan eltávolítunk k-nál kevesebb csúcsot, akkor is még összefüggő marad a gráf. Továbbá lehet k-szorosan élösszefüggő is egy gráf, ha eltávolítunk k-nál kevesebb élt és így is még összefüggő marad a gráf.

3. EU NEMZETKÖZI REPTÉRHÁLÓZAT HIBATŰRŐ KÉPESSÉGÉNEK SZÁMÍTÁSA

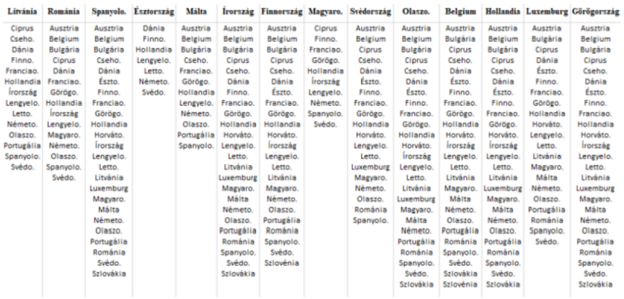
3.1 Adatbázis és lokális él-összefüggőség

A számításhoz elsősorban a többszörös összefüggőséget, mint gráfelméleti elemet fogom használni. Az Európai Unió 27 tagállamánál csak azokat a reptereket vettem figyelembe, amelyeket a tagállamok egyike üzemeltet és nemzetközileg elismert, azaz katonai és magáncélú reptereket szintén kizártam. Az alábbi ábrákon (1-2.ábra) az látható, hogy mely ország melyik másik tagállammal van közvetlen kapcsolatban.



Ausztria	Bulgária	Horvátország	Ciprus	Csehország	Dánia	Németo.	Franciao.	Szlovákia	Portugália	Lengyelo.	Szlovénia	Lettország
Belgium Bulgária Ciprus Dánia Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Luxemburg Németo. Olaszo. Portugália Románia Spanyolo. Svédo.	Ausztria Belgium Ciprus Cseho. Dánia Finno. Franciao. Görögo. Hollandia Írország Lengyelo. Luxemburg Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovákia	Ausztria Belgium Cseho. Dánia Finno. Franciao. Görögo. Hollandia Írország Lengyelo. Letto. Luxemburg Németo. Olaszo. Spanyolo. Svédo. Szlovénia	Ausztria Belgium Bulgária Cseho. Dánia Észto. Finno. Görögo. Hollandia Írország Lengyelo. Letto. Litvánia Magyaro. Németo. Olaszo. Románia	Ausztria Belgium Bulgária Ciprus Dánia Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Letto. Luxemburg Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovákia	Ausztria Belgium Bulgária Ciprus Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Letto. Litvánia Luxemburg Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovénia	Ausztria Belgium Bulgária Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Letto. Litvánia Luxemburg Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovákia Szlovénia	Ausztria Belgium Bulgária Ciprus Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Letto. Litvánia Magyaro. Málta Németo. Olaszo. Spanyolo.	Ausztria Belgium Bulgária Ciprus Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Letto. Litvánia Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo.	Ausztria Belgium Bulgária Ciprus Dánia Észto. Finno. Franciao. Görögo. Hollandia Lengyelo. Németo. Olaszo. Spanyolo.	Dánia Észto. Franciao. Hollandia Litvánia Németo. Olaszo. Svédo.

1. ábra



Litvánia	Románia	Spanyolo.	Észtország	Málta	Írország	Finnország	Magyaro.	Svédország	Olaszo.	Belgium	Hollandia	Luxemburg	Görögország
Ciprus Cseho. Dánia Finno. Franciao. Hollandia Írország Lengyelo. Letto. Németo. Olaszo. Portugália Spanyolo. Svédo.	Ausztria Belgium Bulgária Ciprus Görögo. Hollandia Írország Lengyelo. Magyaro. Németo. Olaszo. Spanyolo. Svédo.	Ausztria Belgium Bulgária Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Letto. Litvánia Luxemburg Magyaro. Málta Németo. Olaszo. Portugália Románia Svédo. Szlovákia	Dánia Finno. Hollandia Lengyelo. Letto. Németo. Svédo.	Ausztria Belgium Bulgária Cseho. Franciao. Görögo. Hollandia Lengyelo. Németo. Olaszo. Portugália Spanyolo.	Ausztria Belgium Bulgária Ciprus Cseho. Dánia Finno. Franciao. Görögo. Hollandia Horvát. Lengyelo. Letto. Litvánia Luxemburg Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovákia	Ausztria Belgium Bulgária Ciprus Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Lengyelo. Letto. Litvánia Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovénia	Ciprus Finno. Franciao. Görögo. Hollandia Írország Lengyelo. Németo. Spanyolo. Svédo.	Ausztria Belgium Ciprus Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Lengyelo. Letto. Luxemburg Magyaro. Málta Németo. Olaszo. Románia Spanyolo.	Ausztria Belgium Bulgária Ciprus Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Letto. Litvánia Luxemburg Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovákia	Ausztria Bulgária Ciprus Cseho. Dánia Észto. Finno. Franciao. Görögo. Hollandia Horvát. Írország Lengyelo. Letto. Litvánia Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovákia Szlovénia	Ausztria Belgium Bulgária Ciprus Cseho. Dánia Észto. Finno. Franciao. Hollandia Horvát. Írország Lengyelo. Letto. Litvánia Luxemburg Magyaro. Málta Németo. Olaszo. Portugália Románia Spanyolo. Svédo. Szlovákia

2. ábra

Az elsődleges célom, megvizsgálni a hálózat csomópontjai közötti kapcsolatot, tehát az él-összefüggőséget kell kiszámolni. Ez az érték megadja a minimum élszámot, amit még eltávolíthatok a gráfból anélkül, hogy szétesne két különböző komponensére. Azonban ehhez a lokális k-él-összefüggőség szükségeltetik, ami Menger tételéből adódóan minden k-élösszefüggő G gráfra igaz, hogy tetszőlegesen kiválasztott két pontja (u, v) között található k darab élidegen út. Ennélfogva egy G gráf lokális él-összefüggőségén az (u, v) csúcsok között elhelyezkedő éldiszjunkt utak maximális számát értjük.

3.2 Gráfok ábrázolási módjai

Mielőtt még a konkrét számításokat végeznénk, fontos megvizsgálni, hogy milyen formában deklarálhatunk egy hálózatot, programozói felületen. A gyakorlatban kétfélét különböztetünk meg: adjacenciamátrix és szomszédsági lista. Az éllistás módszert, ritka (kevés élt tartalmazó) gráfoknál használjuk, hiszen a programban elfoglalt memóriaigénye arányos az élek és pontok számával. Azonban az EU nemzetközi reptérhálózatát tekintve elmondható, hogy a sűrű gráfokhoz tartozik, ezért a mátrix formát választottam.

Ha van egy G= (V, E) gráfunk, aminek a csúcsainak számát n-nel jelöljük, akkor ezt egy n*n-es mátrixba lehet ábrázolni úgy, hogy az oszlop-és sorindexek megegyeznek a gráf csúcsaival. A mátrix feltöltését pedig az alapján adtam meg, hogy adott csúcsoknál van-e él, azaz indul járat a két ország között:

$$C[i,j] = \begin{cases} 1, ha\ (i,j) \in E \\ 0, ha\ (i,j) \notin E \end{cases}$$

Az irányított, súlyozott adjacenciamátrix látható (3.ábra), ahol 0.indexszel jelöltem Ausztriát és a (1-2. ábra) megfelelően haladtam az indexek növelésével egészen 26-os sorszámmal bezárólag, ami Görögországot fogja jelenteni. A sorok jelölik a kiindulási csomópontot, azaz országot, míg az oszlopok a célországot. Az általam létrehozott súlyozási szabályrendszer fő szempontja az, hogy az élkapacitások értékei megegyezzenek az adott kiindulási ország nemzetközi reptereiről indított járatai számával. Vegyük a 0. indexű Ausztriát, akinek 6 nemzetközi reptere van és ezek közül mennyiből lehet eljutni a célországok egyikébe. Érdekességképpen a mátrix átlójában csupa 0 érték került, hiszen ezek a hurokéleknek felelnek meg, azonban a belföldi járatokat nem vettem figyelembe.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0	0	3	2	2	0	2	6	3	0	2	1	0	0	0	1	5	0	0	1	3	0	2	3	2	4	2	6
1	3	0	0	2	2	1	3	3	1	1	3	0	0	0	1	2	0	1	2	1	2	1	2	3	2	2	2
2	4	0	0	0	1	4	7	6	0	0	2	1	2	0	0	2	0	0	2	2	0	3	4	5	6	2	1
3	1	1	0	0	1	1	2	0	0	0	2	0	2	2	1	0	1	0	1	1	1	1	1	1	1	0	2
4	1	4	2	2	0	1	1	1	1	1	2	0	1	0	1	4	0	1	1	1	1	1	2	1	1	1	4
5	3	2	4	2	3	0	4	4	0	4	3	0	2	2	1	6	1	2	3	1	2	3	3	1	2	3	0	5
6	12	13	13	6	3	7	0	12	0	11	10	2	7	6	12	24	4	5	6	6	8	6	12	5	10	7	21
7	4	5	8	0	6	7	8	0	1	20	6	1	2	3	5	15	1	10	14	2	6	7	12	16	15	10	10
8	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	1	0	1	1	0	0	1	1	1	1	0	1
9	1	1	1	0	0	0	4	1	0	0	1	0	0	0	0	3	0	0	1	0	1	0	0	1	1	1	1
10	1	4	5	4	1	5	9	6	0	2	0	0	0	1	1	6	1	5	10	2	3	5	0	2	9	0	6
11	1	1	0	1	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	1
12	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	1	0	0
13	0	0	0	1	1	2	1	1	0	1	1	0	2	0	0	2	0	0	2	1	0	1	1	0	2	0	0
14	5	1	0	3	0	1	8	5	0	0	2	0	0	0	0	6	0	0	3	0	2	1	7	6	3	0	2
15	10	3	1	0	5	4	18	15	1	9	10	0	1	2	8	0	1	4	11	2	5	8	14	17	16	9	2
16	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	2	0	1	0	0	1	0	0
17	1	1	0	0	1	0	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	1
18	3	1	1	1	1	1	4	2	1	5	3	0	1	2	1	5	0	2	0	1	2	1	3	1	2	1	1
19	2	1	1	1	1	1	4	1	0	1	2	1	4	1	1	1	1	0	1	0	1	0	1	5	2	1	1	0
20	0	0	0	1	0	0	2	2	0	0	1	0	0	0	0	1	0	0	1	1	0	1	1	0	0	2	0	1
21	2	0	2	2	2	2	3	2	0	0	4	0	1	0	2	9	1	0	0	2	1	0	1	1	5	1	6
22	12	6	3	2	11	11	27	18	4	12	17	0	6	7	16	21	1	12	12	4	9	9	0	21	20	12	13
23	3	3	3	1	2	2	1	2	1	2	2	2	2	2	2	5	1	2	2	1	2	1	2	0	1	0	4
24	3	2	4	1	2	2	1	4	1	3	2	0	2	2	2	5	1	3	2	2	2	2	4	1	0	1	5
25	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1	0	1
26	15	7	2	4	11	5	19	11	1	1	10	0	3	4	8	6	1	2	6	6	8	9	16	14	14	4	0

3. ábra

3.3 Gráfbejárás kétféle módszerrel

Természetesen minden gráfalgoritmusnál szükségünk lehet a gráf szerkezetében tárolt információkra, csúcsok helyzetére, élkapacitásokra. Ehhez kétféle bejárási stratégiát használhatunk: szélességi (BFS) és mélységi (DFS) algoritmust.

BFS-algoritmus: első lépésként meg kell adni egy kezdőcsomópontot (s), majd megnézi a tőle egy egységnyi távolságra lévő csúcsszomszédjait. Utána az s csúcstól keresi meg a tőle két egységnyi távolságra lévő pontokat, amik egyben s csúcsszomszédjainak a szomszédjai és ez megy tovább mindaddig, még az összes csomópont föl nem lesz derítve. Ha a bejárás közben egyszer már megtaláltunk egy csomópontot, akkor ha újból rálelnénk, nem kell figyelembe venni, azaz kerüljük a duplikációt.

DFS-algoritmus: megadunk egy kezdőcsomópontot, amitől haladunk tovább a véletlenszerűen kiválasztott irányított éleken, mindaddig amíg már nem találunk felderítetlen csúcsot. Még ezzel nem biztos, hogy teljesen föltérképeztünk mindent, így ilyenkor visszamegyünk az út utolsó előtti csúcsáig, majd onnan keresünk tovább haladó útvonalat más bejáratlan csomópontokba. Ha innen sem vezet már újabb út, akkor megint visszamegyünk eggyel korábbi csúcshoz. Végül, ha már nincs több felderítő út a kezdőcsomópontból (s) akkor azt mondhatjuk, hogy s pont mélységi bejárás algoritmusa végéhez ért. Azonban megeshet, hogy kezdőcsomópont bejárása lezárult, de a gráfnak továbbra is marad még felfedezetlen pontja, ez csupán csak annyit jelent, hogy nem minden pontot tudunk elérni a kezdőpontból. Ez esetben választani kell egy új kiindulási pontot (felfedezetlen pontok közül), majd tovább folytatjuk a DFS-algoritmust.

3.3 Ford-Fulkerson tétel

Ahhoz, hogy a hálózatnak kiszámítsuk a lokális él-összefüggőségét, szükségünk van a Ford-Fulkerson tételére, ami a maximális folyam és minimális vágás elvén működik. Hálózatnak tekintjük az összes olyan G gráfot, melynek két kitüntetett pontja van forrás (s) és cél (t) és minden élhez hozzárendeltünk egy nemnegatív valós számot, ami az él kapacitását jelenti (c(e)), összefoglalva: egy hálózat (G; s; t; c) összetevői.

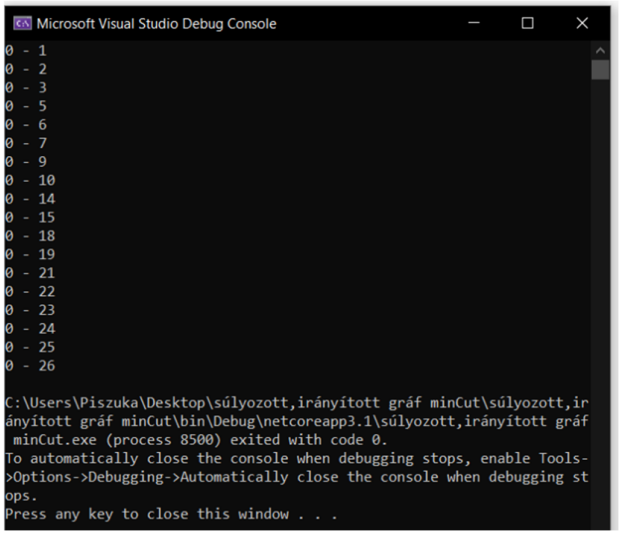
Folyamnak nevezzük azt az f függvényt, mely minden élre igaz, hogy f(e) ≤ c(e) és $\sum\{f(e)|e\ végpontja\ v\} - \sum\{f(e)|e\ kezdőpontja\ v\} = 0$ és $v \in V(G)$ kivéve az s, t pontokat, tehát egy csúcsba befolyó folyamértékek összege mindig megegyezik a csúcsból kifolyóval. Gyakorlatban törekszünk minél nagyobb kihasználtságra egy-egy hálózat esetében, erre használunk javító gráfokat, amik megvizsgálják, hogyan lehetne növelni a folyam értékét.

Legyen $X \subseteq V(G)$ a csúcsoknak egy nem üres részhalmaza, és tegyük fel, hogy s ∈ X, valamint t ∈ V(G)-X. Az olyan éleket, amik X halmazból V(G)-X halmazba mutatnak nevezzük (s,t)-vágásnak. Ez a vágás meggátolja az átjutást s-ből t-be. A vágás kapacitását (c(C)) a vágásban szereplő éleken lévő kapacitások összege határozza meg, amely X részhalmaz egyik pontjából a V(G)-X részhalmaz egy másik pontjába mutatnak, de csak az előremutató éleket vesszük figyelembe.

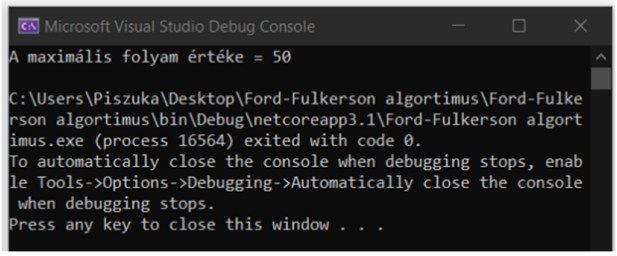
Végezetül a Ford-Fulkerson tételére térnék át, amely magába foglalja azt, hogy egy hálózatban felfedezett maximális folyam értéke megegyezik a minimális vágás értékével és egyben megszünteti az s, t pontok közti kapcsolatot.

4. IMPLEMENTÁCIÓS EREDMÉNYEK ELEMZÉSE

Miután átvettük a szükséges gráfalgoritmusokat, nézzük meg, milyen eredményekre tehetünk szert, mindezt C# programnyelven keresztül. Ügyelve a gráf megfelelő bementi formájának deklarálására, majd a gráfbejárási metódusokat követi a minimális vágást kereső algoritmus, végezetül pedig a kiíratás. A 4. ábra mutatja, hogy Ausztriának, mint kiindulási csomópont, mely élei tartoznak a minimális vágáshoz szükséges halmazba tehát, ha egyidejűleg ezeket az éleket eltávolítanám, akkor Ausztria izolált ponttá válna, elszigetelődne a többi országtól. Az 5. ábra pedig a minimális vágáshoz tartozó élek kapacitásának összegét adja, ami egyben a maximális folyamértékével is megegyezik. Az eredményt tekintve elmondható, hogy Ausztria nemzetközi reptereiről indított 50 járatot törölni kell, hogy megszűnjön a kapcsolat a többi tagállammal. Továbbá Ausztria sem 100%-osan védett a támadásokkal szemben, hiszen a 22 lokális él-összefüggőségét növelni kellene még új élekkel, hogy legalább minden tagállammal legyen egy közös élük.

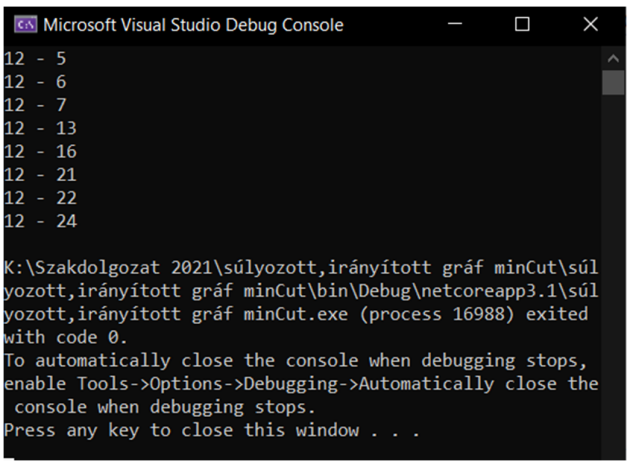


4. ábra

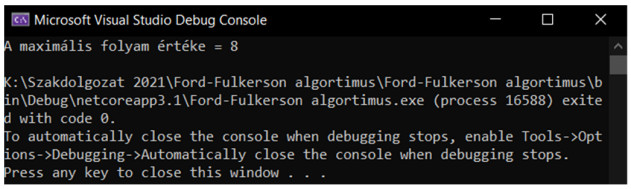


5. ábra

Összehasonlításképpen megnéztem, melyik ország rendelkezik a legkisebb hibatűrő képességgel. Az egyik Lettország, akinek az alábbi (6.ábra) látszik, mely éleket kellene eltávolítani és ez egyuttal 8 járat törlésével járna (7.ábra).



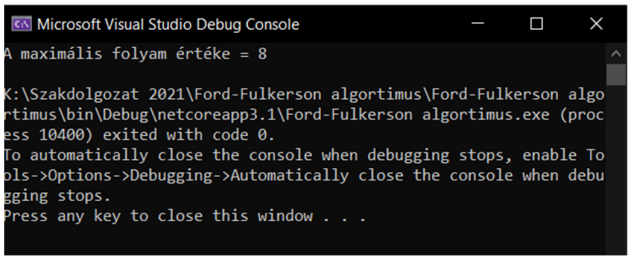
6. ábra



7. ábra

Észtország szintén az alacsony hibatűrő képességű országok közé tartozik, hiszen az alábbi ábrákon (8-9.ábra) látszik, hogy csupán 7 élt kell eltávolítani vele együtt 8 járatot, hogy a hálózat két komponensére essen szét és Észtország elveszítse kapcsolatát a többi tagállammal.

8. ábra



9. ábra

5. MILYEN HATÁSSAL BÍR EGY-EGY ÉL ELTÁVOLÍTÁS A HÁLÓZATRA NÉZVE

Elsődleges szempont, hogy megvizsgáljuk, milyen károkat tud okozni egy-egy él eltávolítása a reptérhálózatban. Ahogy említettem Ausztria 22 tagállammal nemzetközileg közvetlen kapcsolatban áll, de így is elmondható, hogy nagy veszteségeket tud kiváltani egy-egy kibertámadás vagy természeticsapás. A támadott ország infrastruktúrájának kármértéke függ a blokkolás időtartamától. Azonban már rövid időn belül olyan zavart tud létrehozni az ütemezésben, hogy az komoly gazdasági és logisztikai megoldásokat igényelne.

Ugyanakkor mégis előfordul a gyakorlatban, hogy lokális hibák miatt egyes csomópontok nem tudják elvégezni a feladataikat, ilyenkor továbbadják más pontoknak, akik szintén továbbadják a rájuk nehezedő feladatokat, ezzel elindul egy lavinahatás. Viszont már kifejlesztettek különböző optimalizáló algoritmusokat, hogy egyenletesen szétoszlassák a terhelést a szomszédos csomópontok között. Erre példaként szolgálhat olyan logisztikai protokoll, aminek segítségével megtalálhatjuk a hálózatban az alternatív útvonalat, ami a hibaelhárításáig segít a hálózatot összefüggőként kezelni. Ehhez az INCOTERMS 2020-as szokványt, klauzulákat használják vagy egy-egy alkatrész beszerzését megpróbálják nem csak egy országból megrendelni. Azonban ez készlet felhalmozódással járhat, amit próbálnak elkerülni a megrendelő cégek. Miután megoldódott a hiba az adott csomópontnál, visszaállhat a hálózatba és tovább folytathatja a feladatát.

6. KONKLÚZIÓ

Ebben a cikkben megvizsgáltam az Európai Unió tagállamaiból álló reptérhálózatának rezisztenciáját támadásokkal szemben, ehhez gráf-és folyamalgoritmusokat használtam. Ezen belül az egyes országok közti követlen kapcsolatot vettem figyelembe. Megkerestem a legideálisabb gráfábrázolási módot, amivel könnyedén bedeklaráltam C# programnyelv környezetébe. Végezetül az ellenőrzés sem maradhatott el, amihez a Ford-Fulkerson tételt használtam, továbbá ok-okozati összefüggésekre kerestem optimalizálási módszereket.

HIVATKOZÁSOK

Katona Y. Gy., Recski A., Szabó Cs. (2002) *A számítástudomány alapjai*, Typotex kiadó, Budapest

Lovász L., Pelikán J., Vesztergombi K. (2019). *Diszkrét Matematika*, Typotex kiadó, Budapest

Oláh K. (2021). *Gráfok többszörös összefüggősége,* Óbudai Egyetem, Budapest

Zentai D. (2017) *Gráfelméleti módszerek a kritikus infrastruktúra védelemben.* Hadmérnök XII. évfolyam 2.pp.343-344.