Gráfelméleti heurisztikák alkalmazása hibatűrő hálózatok tervezésénél

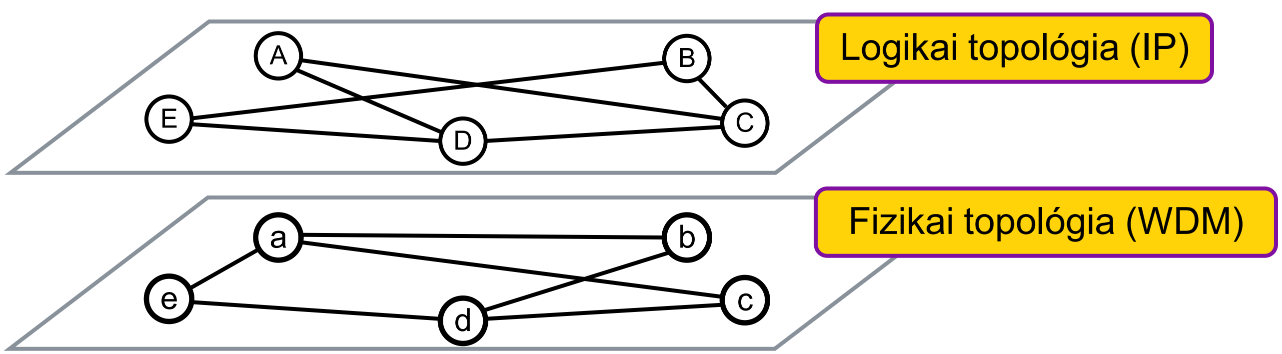
Radics Norbert

Nokia Siemens Networks

Hálózattervezés

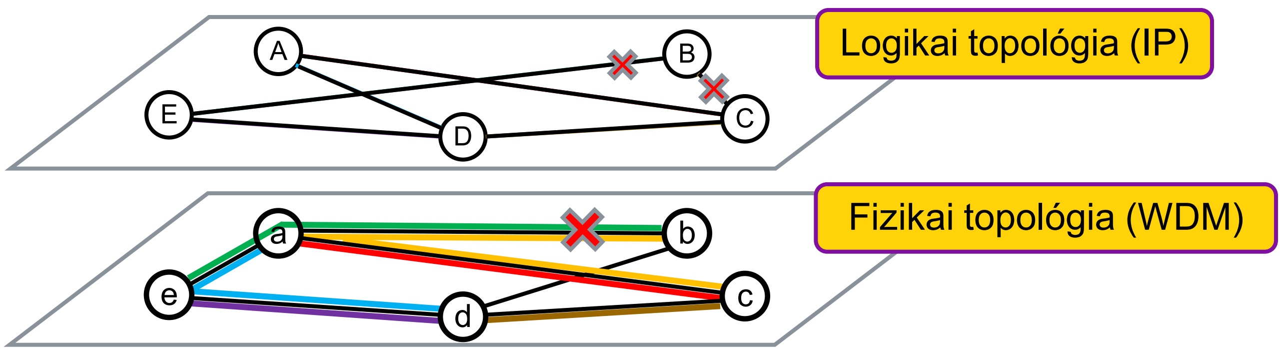
A többrétegű gerinchálózatok tervezése, mint mérnöki feladat főbb összetevői a következők:

- Fizikai topológia tervezése (optikai hálózat),
- Logikai topológia/topológiák tervezése (pl. IP hálózat)
- A topológiák közti leképezés (mapping), azaz hogy a logikai topológia élei által meghatározott adatátvitelt mely fizikai éleket (fizikai hálózati eszközöket) használva valósítjuk meg.



Hibatűrő hálózatok – Survivable Mapping

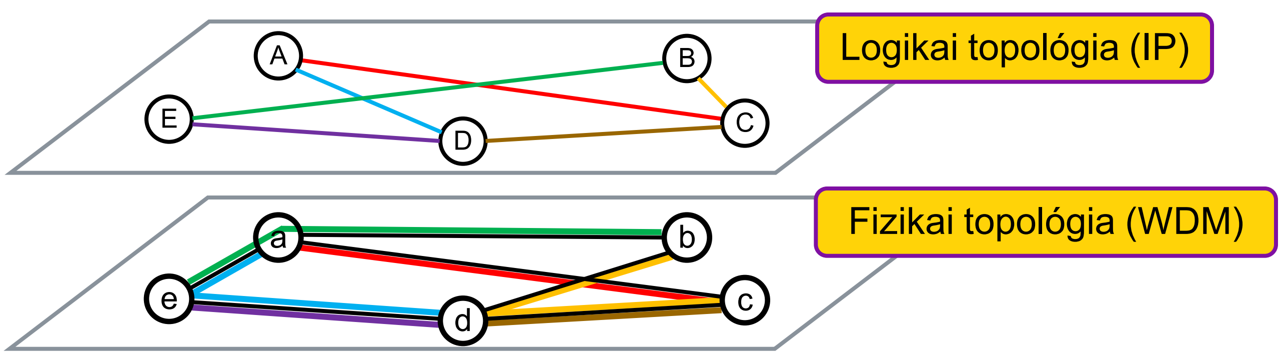
- A topológiák közti leképezés (mapping) során minden logikai élnek megfeleltetünk a fizikai hálózatban egy, az adott él végpontjai közt vezető utat.
- Mivel egy adott fizikai él (optikai kábel) több logikai él útjában is szerepelhet, egy fizikai él meghibásodása rendszerint több logikai élen akadályozza meg az adatátvitelt.
- Hibatűrő hálózatoknak rendelkezniük kell egy helyreállító mechanizmussal, ami a logikai rétegben a meghibásodott élek forgalmát alternatív utakra irányítja át (IP restoration). Ennek alapfeltétele a topológiák közti leképezés ama tulajdonsága, hogy a fizikai réteg (előre meghatározott) hibáinak esetén a logikai topológia „összefüggő” maradjon, azaz bármely két csomópont közt megvalósítható maradjon az adatátvitel. Az ilyen leképezéseket hívjuk *survivable mapping*-nek.



Nem survivable mapping

Hibatűrő hálózatok – Survivable Mapping

- A topológiák közti leképezés (mapping) során minden logikai élnek megfeleltetünk a fizikai hálózatban egy, az adott él végpontjai közt vezető utat.
- Mivel egy adott fizikai él (optikai kábel) több logikai él útjában is szerepelhet, egy fizikai él meghibásodása rendszerint több logikai élen akadályozza meg az adatátvitelt.
- Hibatűrő hálózatoknak rendelkezniük kell egy helyreállító mechanizmussal, ami a logikai rétegben a meghibásodott élek forgalmát alternatív utakra irányítja át (IP restoration). Ennek alapfeltétele a topológiák közti leképezés ama tulajdonsága, hogy a fizikai réteg (előre meghatározott) hibáinak esetén a logikai topológia „összefüggő” maradjon, azaz bármely két csomópont közt megvalósítható maradjon az adatátvitel. Az ilyen leképezéseket hívjuk *survivable mapping*-nek.



Survivable mapping

A kiinduló probléma és annak bonyolultsága

PROBLÉMA (**SM**): adott 2-rétegű hálózat (azaz adott logikai és fizikai topológiák) esetén létezik-e survivable mapping a topológiák között az egyszeres élhibák kivédésére?

Állítás: **SM** NP-teljes

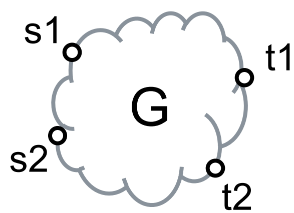
1. **SM** $\in$ NP $\rightarrow$ fizikai topológia éleit egyenként elhagyva összefüggőség ellenőrzése

A kiinduló probléma és annak bonyolultsága

PROBLÉMA (**SM**): adott 2-rétegű hálózat (azaz adott logikai és fizikai topológiák) esetén létezik-e survivable mapping a topológiák között az egyszeres élhibák kivédésére?

<u>Állítás:</u> **SM** NP-teljes

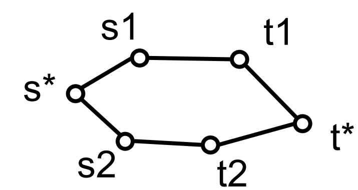
2. Többtermékes egészértékű folyamprobléma visszavezethető **SM**-re



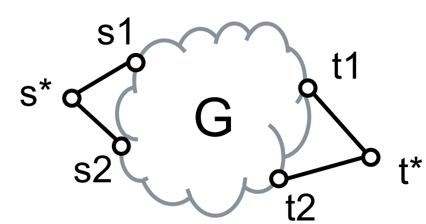
$s1 \rightarrow t1$ folyamérték = 1
$s2 \rightarrow t2$ folyamérték = 1
$\forall$ élkapacitás = 1

$\longrightarrow$

logikai topológia



fizikai topológia



$\exists$ két-termékes folyam $\Leftrightarrow$ $\exists$ survivable mapping

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ①

O. Crochat, J.-Y. Le Boudec :

“Design Protection for WDM Optical Networks”

IEEE J. Select. Areas Commun., Vol.16, No.7, pp.1158-1165, September 1998

Bemenet:

- logikai topológia (kétszeresen él-összefüggő), egységnyi forgalom igénnyel minden élen
- fizikai topológia (kétszeresen él-összefüggő), kapacitás korláttal az éleken)

Kimenet:

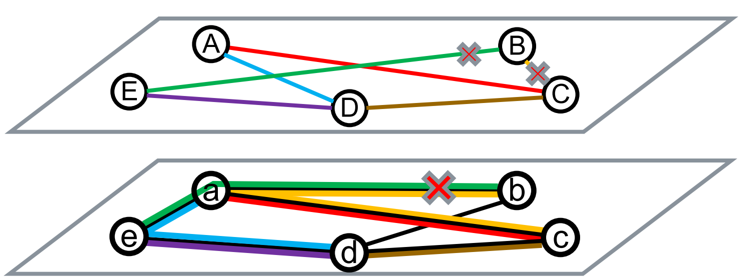
- a logikai topológia éleinek leképezése a fizikai topológiára, minimalizálva a kritikus élek (melyek hibája a logikai topológia több komponensre való szétesését okozza) számát

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ① - a modell

- Adott paraméterek
  - N : csomópontok száma a hálózatban
  - $L=(l_{mn})$: a fizikai topológia $N \times N$ -es szomszédsági mátrixa, $l_{mn} = l_{nm} =$ az adott fizikai él kapacitása
  - $C=(c_{sd})$: a logikai topológia $N \times N$ -es szomszédsági mátrixa, $c_{sd} = 1 \Leftrightarrow$ sd él létezik, $c_{sd} = 0$ egyébként
- Alap változók
  - $R_{sd} = (r_{mn}^{sd})$: a topológiák közti leképezés $N \times N$-es mátrixa, $r_{mn}^{sd} = r_{nm}^{sd} = 1 \Leftrightarrow$ *SD* élhez tartozó út áthalad az *mn* élen
- További változók
  - $b_{mn}^{sd} = 1 \Leftrightarrow$ ha az *mn* fizikai él hibája esetén az *sd* logikai él meghibásodik és nem váltható ki alternatív úttal $\rightarrow$ az {*sd*, *mn*} párt „*törékeny pár*”-nak hívjuk (an *mn* él pedig *kritikus* él)

$b_{ab}^{BC} = 1$

$b_{ab}^{BE} = 1$



- Feltételek
  - Kapacitás feltételek
  $$\sum_{s,d} r_{mn}^{sd} \le l_{mn}$$
  - Routing feltételek
  $$\sum_{m} r_{mn}^{sd} = \begin{cases} 1, & \text{ha } n = s \text{ or } d \\ 2, & \text{ha } n \text{ rajta van az } sd\text{-hez tartozó úton} \\ 0, & \text{egyébként} \end{cases}$$
- Célfüggvény
  min F
  - Minimalizáljuk a *törékeny párok* számát
  $$F = \sum_{m,n} \sum_{s,d} b_{mn}^{sd}$$
  - (alternatív függvény)
  minimalizáljuk az egy adott hiba esetén érintett logikai élek maximális számát:
  $$F = \max\left(\sum_{s,d} b_{mn}^{sd}\right)$$

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ① - heurisztikus megoldási módszer

DAP algoritmus (Disjoint Alternate Path)

1. Kiindulunk egy tetszőleges megoldásból;
2. Véletlen módon változtatjuk a logikai élekhez tartozó utakat, csökkentve a *törékeny párok* keletkezésének valószínűségét;
3. Kiválasztjuk a megoldást amely minimalizálja az *F* célfüggvényt;
4. Az *R* mátrix alapján újraszámoljuk a kritikus élek és a törékeny párok listáját;
5. Folytatjuk az iterációt a 2. lépéssel.

Utak módosítása:

Az éleket véletlen sorrendben véve módosítjuk a hozzájuk rendelt fizikai utakat, tekintetbe véve az alábbi „tabu listákat” :

- **Tabu élek:** ebben tároljuk a logikai éleket melyek leképezését legutóbb változtattunk.
- **Tabu lépések:** logikai élenkénti lista, melyeken az élhez rendel korábbi utak egy-egy (lehetőleg kritikus) élét tároljuk. Ha nem találunk utat egy módosító lépésnél, a legkorábbi élt elhagyjuk a tabu lépések közül.

Kilépési feltétel:

Iterációk száma *F* értékének csökkenése nélkül > *max_iteráció_szám*

VAGY

F = 0

Kapacitás korlát figyelembe vétele:

Az útkeresés során nem törődünk a kapacitás korlátokkal. Ezért „cserébe” módosítjuk a célfüggvényt, így szorítva le a fizikai topológia élein a forgalmat a kapacitás korlát alá:

$F_{calc}=F+F_C$, ahol

$$F_C = \sum_{m,n}\left[\max\left(0, \sum_{s,d} r_{mn}^{sd} - l_{mn}\right)\right]^2$$

Nokia Siemens Networks

Egészértékű Lineáris Programozás megközelítés

A probléma ELP modellje ②

A. Nucci, B. Sansò, T. G. Crainic, E. Leonardi, M. A. Marsan :
"On the Design of Fault-Tolerant Logical Topologies in Wavelength-Routed Packet Networks"
IEEE J. Select. Areas Commun., Vol.22, No.9, pp.1884-1895, November 2004

Bemenet:

- fizikai topológia (kétszeresen él-összefüggő), kapacitás korláttal az éleken
- fokszám korlát a logikai topológia csúcsaira (korlátozott számú receiver-ek és transmitter-ek)
- logikai topológia csúcs-párjai között megkövetelt forgalom
- a védeni kívánt fizikai él-hibák (alapesetben az összes egyszeres él-hiba)

Kimenet:

- egy logikai topológia és a logikai élek leképezése a fizikai topológiára, melyek optimalizálják a megfelelő célfüggvényt;

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - a modell

ELP megfogalmazása a problémának

- paraméterek
  - N : csúcspontok száma
  - $G^0 = (V, E^0)$: fizikai topológia irányított gráfja
  - R_i, T_i : receiverek/transmitterek száma az $i \in V$ csúcsban
  - S_v : a $v \in E^0$ él meghibásodásához tartozó hálózati állapot
  - S_0 : a hibamentes hálózati állapot
  - $\mathcal{E}$: a logikai topológia lehetséges éleinek halmaza
  - $G^1(S_v) = (V, E^1(S_v))$: logikai gráf a hálózat S_v állapotában, amelyet a $G^1(S_0)$ gráfból kapunk azon élek elhagyásával, melyekhez rendelt út áthalad a meghibásodott $v \in E^0$ élen.
  - $\Lambda = (\lambda_{sd})$: forgalom-mátrix
- változók
  - X_u : =1 ha az $u \in \mathcal{E}$ logikai él eleme a $G^1(S_0)$ logikai topológiának; =0 egyébként
  - Y_{uv} : =1 ha az $u \in \mathcal{E}$ élhez tartozó út áthalad a $v \in E^0$ fizikai élen; =0 egyébként
  - $T_u^{sd}(S_v)$: = λ_{sd} ha az s→d forgalom áthalad az $u \in \mathcal{E}$ logikai élen az S_v állapotban; =0 egyébként

- feltételek
  - fokszám korlát

$$\sum_{u \in \Gamma^+(i)} X_u \le T_i \quad \forall i \in V \qquad\qquad \sum_{u \in \Gamma^-(i)} X_u \le R_i \quad \forall i \in V$$

Egy adott csúcshoz kapcsolódó logikai linkek száma nem haladhatja meg a rendelkezésre álló receiverek illetve transmitterek számát.

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - a modell

ELP megfogalmazása a problémának

- paraméterek
  - N : csúcspontok száma
  - $G^0 = (V, E^0)$: fizikai topológia irányított gráfja
  - R_i, T_i : receiverek/transmitterek száma az $i \in V$ csúcsban
  - S_v : a $v \in E^0$ él meghibásodásához tartozó hálózati állapot
  - S_0 : a hibamentes hálózati állapot
  - $\mathcal{E}$: a logikai topológia lehetséges éleinek halmaza
  - $G^1(S_v) = (V, E^1(S_v))$: logikai gráf a hálózat S_v állapotában, amelyet a $G^1(S_0)$ gráfból kapunk azon élek elhagyásával, melyekhez rendelt út áthalad a meghibásodott $v \in E^0$ élen.
  - $\Lambda=(\lambda_{sd})$: forgalom-mátrix

- változók
  - X_u : =1 ha az $u \in \mathcal{E}$ logikai él eleme a $G^1(S_0)$ logikai topológiának; =0 egyébként
  - Y_{uv} : =1 ha az $u \in \mathcal{E}$ élhez tartozó út áthalad a $v \in E^0$ fizikai élen; =0 egyébként
  - $T_u^{sd}(S_v)$: = λ_{sd} ha az s→d forgalom áthalad az $u \in \mathcal{E}$ logikai élen az S_v állapotban; =0 egyébként

- feltételek
  - fokszám korlát

$$\sum_{u\in\Gamma^+(i)} X_u \le T_i \quad \forall i \in V \qquad\qquad \sum_{u\in\Gamma^-(i)} X_u \le R_i \quad \forall i \in V$$

  - routing

$$\sum_{u\in\Gamma^+(i)} t_u^{sd}(S_v) - \sum_{u\in\Gamma^-(i)} t_u^{sd}(S_v) = \begin{cases} \lambda_{sd}, & \text{ha } s=i \\ -\lambda_{sd}, & \text{ha } d=i \\ 0, & \text{egyébként} \end{cases}$$

$$\forall s,d,i \in V,\ \forall S_v$$

$$t_u^{sd}(S_v) \le X_u \lambda_{sd} \qquad \forall s,d \in V,\ \forall S_v,\ \forall u \in \mathcal{E}$$

Egy adott élen áthaladó forgalom nem haladhatja meg az él kapacitását, és nem haladhat forgalom egy élen, ha az él nem eleme a logikai topológiának.

Bemenő/kijövő forgalom ellenőrzése csúcsonként

Nokia Siemens Networks

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - a modell

ELP megfogalmazása a problémának

- paraméterek
  - N : csúcspontok száma
  - $G^0 = (V, E^0)$: fizikai topológia irányított gráfja
  - R_i, T_i : receiverek/transmitterek száma az $i \in V$ csúcsban
  - S_v : a $v \in E^0$ él meghibásodásához tartozó hálózati állapot
  - S_0 : a hibamentes hálózati állapot
  - $\mathcal{E}$: a logikai topológia lehetséges éleinek halmaza
  - $G^1(S_v) = (V, E^1(S_v))$: logikai gráf a hálózat S_v állapotában, amelyet a $G^1(S_0)$ gráfból kapunk azon élek elhagyásával, melyekhez rendelt útvonal tartalmazza a meghibásodott $v \in E^0$ élen.
  - $\Lambda = (\lambda_{sd})$: forgalom-mátrix
- változók
  - X_u : =1 ha az $u \in \mathcal{E}$ logikai él eleme a $G^1(S_0)$ logikai topológiának; =0 egyébként
  - Y_{uv} : =1 ha az $u \in \mathcal{E}$ élhez tartozó útvonal átmegy a v fizikai élen; =0 egyébként
  - $T_u^{sd}(S_v)$: = λ_{sd} ha az s→d forgalom áthalad az $u \in \mathcal{E}$ logikai élen az S_v állapotban; =0 egyébként

- feltételek
  - fokszám korlát

$$\sum_{u \in \Gamma^+(i)} X_u \le T_i \quad \forall i \in V \qquad\qquad \sum_{u \in \Gamma^-(i)} X_u \le R_i \quad \forall i \in V$$

  - routing

$$\sum_{u \in \Gamma^+(i)} t_u^{sd}(S_v) - \sum_{u \in \Gamma^-(i)} t_u^{sd}(S_v) = \begin{cases} \lambda_{sd}, & \text{ha } s = i \\ -\lambda_{sd}, & \text{ha } d = i \\ 0, & \text{egyébként} \end{cases}$$

$$\forall s, d, i \in V, \ \forall S_v$$

$$t_u^{sd}(S_v) \le X_u \lambda_{sd} \qquad \forall s, d \in V, \ \forall S_v, \ \forall u \in \mathcal{E}$$

  - Leképezés (mapping)

$$Y_{uv} \le X_u \quad \forall u \in \mathcal{E}, v \in E^0$$

$$\sum_{v \in \Theta^+(i)} Y_{uv} - \sum_{v \in \Theta^-(i)} Y_{uv} = \begin{cases} 1, & \text{ha } Origin(u) = i \\ -1, & \text{ha } Destination(u) = i \\ 0, & \text{egyébként} \end{cases}$$

$$\forall u \in \mathcal{E}, \forall i \in V$$

$$\sum_{s,d \in V} t_u^{sd}(S_v) \le \left(\sum_{s,d \in V} \lambda_{sd} \right) (1 - Y_{uv})$$

$$\forall u \in \mathcal{E}, \ \forall v \in E^0, \ \forall S_v \ne S_0$$

Csak a topológiában szereplő éleket képezzük le

Utak ellenőrzése

Ha az adott állapotban a fizikai él hibás, nem haladhat forgalom az azt használó logikai éleken

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - a modell

ELP megfogalmazása a problémának

- paraméterek
  - N : csúcspontok száma
  - $G^0 = (V, E^0)$: fizikai topológia irányított gráfja
  - R_i, T_i : receiverek/transmitterek száma az $i \in V$ csúcsban
  - S_v : a $v \in E^0$ él meghibásodásához tartozó hálózati állapot
  - S_0 : a hibamentes hálózati állapot
  - $\mathcal{E}$: a logikai topológia lehetséges éleinek halmaza
  - $G^1(S_v) = (V, E^1(S_v))$: logikai gráf a hálózat S_v állapotában, amelyet a $G^1(S_0)$ gráfból kapunk azon élek elhagyásával, melyekhez rendelt út áthalad a meghibásodott $v \in E^0$ élen.
  - $\Lambda = (\lambda_{sd})$: forgalom-mátrix

- változók
  - X_u : =1 ha az $u \in \mathcal{E}$ logikai él eleme a $G^1(S_0)$ logikai topológiának; =0 egyébként
  - Y_{uv} : =1 ha az $u \in \mathcal{E}$ élhez tartozó út áthalad a $v \in E^0$ fizikai élen; =0 egyébként
  - $T_u^{sd}(S_v)$: = λ_{sd} ha az s→d forgalom áthalad az $u \in \mathcal{E}$ logikai élen az S_v állapotban; =0 egyébként

- feltételek
  - fokszám korlát

$$\sum_{u \in \Gamma^+(i)} X_u \le T_i \quad \forall i \in V \qquad \sum_{u \in \Gamma^-(i)} X_u \le R_i \quad \forall i \in V$$

  - routing

$$\sum_{u \in \Gamma^+(i)} t_u^{sd}(S_v) - \sum_{u \in \Gamma^-(i)} t_u^{sd}(S_v) = \begin{cases} \lambda_{sd}, & \text{ha } s = i \\ -\lambda_{sd}, & \text{ha } d = i \\ 0, & \text{egyébként} \end{cases}$$

$$\forall s,d,i \in V, \ \forall S_v$$

$$t_u^{sd}(S_v) \le X_u \lambda_{sd} \qquad \forall s,d \in V, \ \forall S_v, \ \forall u \in \mathcal{E}$$

  - Leképezés (mapping)

$$Y_{uv} \le X_u \quad \forall u \in \mathcal{E}, v \in E^0$$

$$\sum_{v \in \Theta^+(i)} Y_{uv} - \sum_{v \in \Theta^-(i)} Y_{uv} = \begin{cases} 1, & \text{ha } Origin(u) = i \\ -1, & \text{ha } Destination(u) = i \\ 0, & \text{egyébként} \end{cases}$$

$$\forall u \in \mathcal{E}, \forall i \in V$$

$$\sum_{s,d \in V} t_u^{sd}(S_v) \le \left(\sum_{s,d \in V} \lambda_{sd} \right) (1 - Y_{uv})$$

$$\forall u \in \mathcal{E}, \ \forall v \in E^0, \ \forall S_v \ne S_0$$

  - fizikai élek kapacitás korlátja

$$\sum_{u \in \mathcal{E}} Y_{uv} \le \text{max kapacitás}$$

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - a modell

ELP megfogalmazása a problémának

- paraméterek
  - N : csúcspontok száma
  - $G^0 = (V, E^0)$: fizikai topológia irányított gráfja
  - R_i, T_i : receiverek/transmitterek száma az $i \in V$ csúcsban
  - S_v : a $v \in E^0$ él meghibásodásához tartozó hálózati állapot
  - S_0 : a hibamentes hálózati állapot
  - $\mathcal{E}$: a logikai topológia lehetséges éleinek halmaza
  - $G^1(S_v) = (V, E^1(S_v))$: logikai gráf a hálózat S_v állapotában, amelyet a $G^1(S_0)$ gráfból kapunk azon élek elhagyásával, melyekhez rendelt út áthalad a meghibásodott $v \in E^0$ élen.
  - $\Lambda=(\lambda_{sd})$: forgalom-mátrix

- változók
  - X_u : =1 ha az $u \in \mathcal{E}$ logikai él eleme a $G^1(S_0)$ logikai topológiának; =0 egyébként
  - Y_{uv} : =1 ha az $u \in \mathcal{E}$ élhez tartozó út áthalad a $v \in E^0$ fizikai élen; =0 egyébként
  - $T_u^{sd}(S_v)$: = λ_{sd} ha az s→d forgalom áthalad az $u \in \mathcal{E}$ logikai élen az S_v állapotban; =0 egyébként

- célfüggvény

$$\min H$$

ahol

$$H \geq \left[\sum_{s,d \in V} t_u^{sd}(S_v) \right] \qquad \forall S_v,\ \forall u \in \mathcal{E}$$

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - heurisztikus megoldási módszer

Tabu Search módszer lépései

- Kiindulási megoldás
- Módosítások, szomszédos megoldások
- Megoldások értékelése
- Kilépési feltétel

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - heurisztikus megoldási módszer

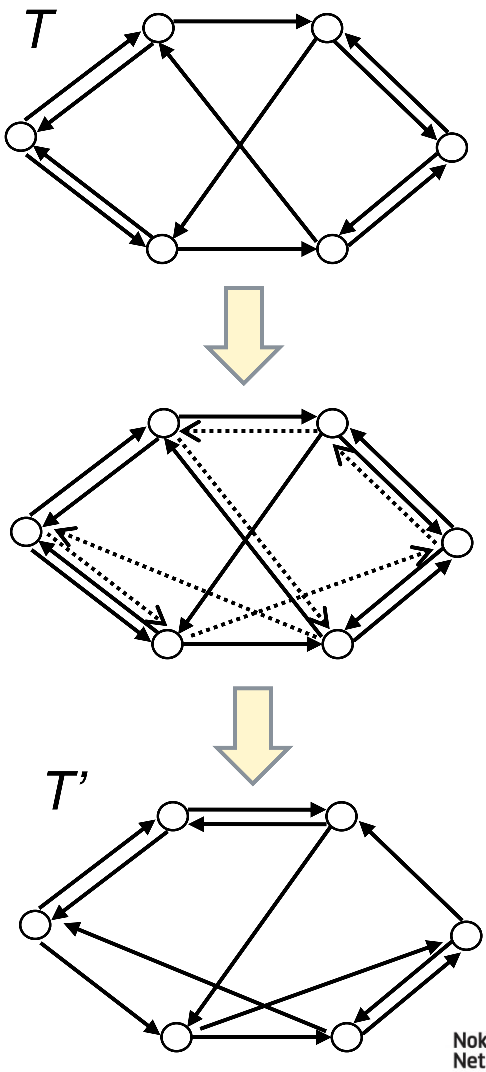
Tabu Search módszer lépései

- Kiindulási megoldás
- Módosítások, szomszédos megoldások

  N(T) -t, a *T* logikai topológia szomszédait úgy kapjuk meg *T*-ből, hogy adott hosszúságú „alternáló körök” mentén kicseréljük az éleket.

  Az alternáló kör páratlan élei a topológia egy élével haladnak szembe, míg a páros élek ott haladnak, ahol nincs él a topológiában az adott irányban.

  T'-t úgy kapjuk, hogy az alternáló kör páratlan éleinél a szembe menő éleket elhagyjuk a topológiából, míg a páros élekkel megegyező irányú új éleket hozzáadunk.
- Megoldások értékelése
- Kilépési feltétel



Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - heurisztikus megoldási módszer

Tabu Search módszer lépései

- Kiindulási megoldás
- Módosítások, szomszédos megoldások

  N(T) -t, a *T* logikai topológia szomszédait úgy kapjuk meg *T*-ből, hogy adott hosszúságú „alternáló körök” mentén kicseréljük az éleket.
- Megoldások értékelése

  Az *N(T)*-ben lévő topológiák éleit leképezzük a fizikai topológiára. Utána, minden egyes hálózat-állapotra a szükséges forgalmat átvezetjük a topológia élein, majd kiértékeljük a célfüggvényt.
- Kilépési feltétel

OR(i), *IR(i)* : a már leképezett logikai linkek halmaza, melyek az *i* csúcsból indulnak, illetve oda érkeznek.

ON(i), *IN(i)* : a még le nem képezett linkek halmazai.

V_{ij} : logikai él melynek végpontjai *i* és *j*.

Q : csúcsok egy halmaza. Kiindulásként a hálózat összes csúcsát tartalmazza.

GDAP algoritmus (Greedy Disjoint Alternate Path):

Step 0: Azon V_{ij} logikai éleket, melyek végpontjai a fizikai topológiában is össze vannak kötve, erre az egy-élű útra képezzük. V_{ij} –t betesszük az *OR(i)*, *IR(j)* halmazokba, és elhagyjuk az *ON(i)*, *IN(j)* halmazokból.

Step 1: Ha $Q=\varnothing$, STOP, egyébként pedig véletlenül kiválasztunk egy *i* csúcsot, és elhagyjuk *Q-ból*.

Step 2: Ha $ON(i)=\varnothing$, GOTO Step 3, egyébként véletlen sorrendben sorra vesszük a $V_{ik} \in ON(i)$ éleket, és keresünk hozzá egy legrövidebb utat a fizikai topológiában amely éldiszjunkt az *OR(i)* –ben lévő V_{ij} –k útjaitól, illetve az *IR(k)*-ban lévő V_{jk} –k útjaitól. Ha nem létezik ilyen legrövidebb út, akkor V_{ik} –t a legrövidebb útra képezzük le. Ha az sem létezik, akkor V_{ik} –t nem képezzük le.

Step 3: Ha $IN(i)=\varnothing$, GOTO Step 1, egyébként véletlen sorrendben sorra vesszük a $V_{ki} \in IN(i)$ éleket, és keresünk hozzá egy legrövidebb utat a fizikai topológiában amely éldiszjunkt az *IR(i)* –ben lévő V_{ji} –k útjaitól, illetve az *OR(k)*-ban lévő V_{kj} –k útjaitól. Ha nem létezik ilyen legrövidebb út, akkor V_{ki} –t a legrövidebb útra képezzük le. Ha az sem létezik, akkor V_{ki} –t nem képezzük le.

Egészértékű Lineáris Programozás megközelítés
A probléma ELP modellje ② - heurisztikus megoldási módszer

Tabu Search módszer lépései

- Kiindulási megoldás
- Módosítások, szomszédos megoldások

  N(T) -t, a *T* logikai topológia szomszédait úgy kapjuk meg *T*-ből, hogy adott hosszúságú „alternáló körök” mentén kicseréljük az éleket.
- Megoldások értékelése

  Az *N(T)*-ben lévő topológiák éleit leképezzük a fizikai topológiára. Utána, minden egyes hálózat-állapotra a szükséges forgalmat átvezetjük a topológia élein, majd kiértékeljük a célfüggvényt.
- Kilépési feltétel

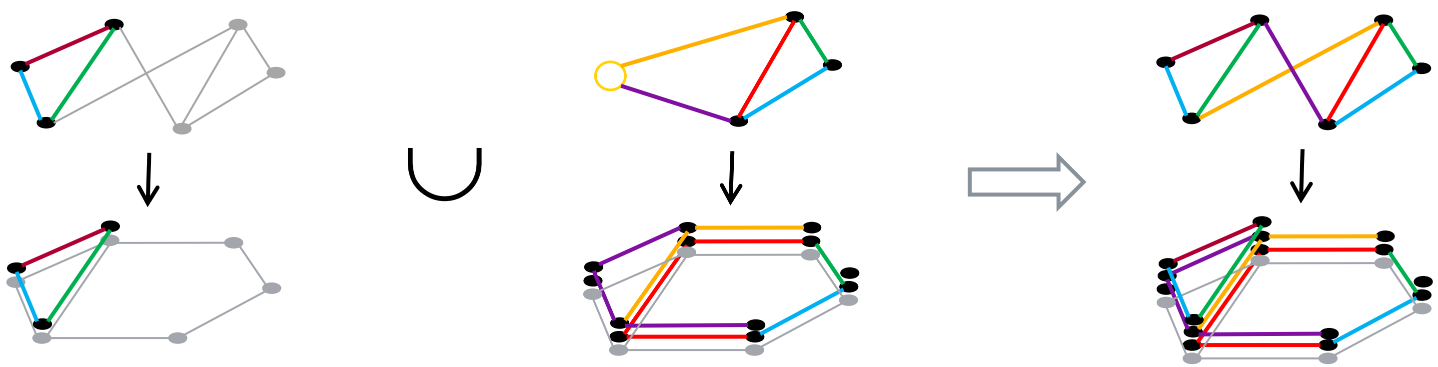
  A tabu search eljárás befejeződik, ha az iterációk száma meghaladja a (hálózat méretétől illetve a megoldás megkövetelt minőségétől függő) maximális iteráció-számot.

Gráf-elméleti megközelítés
Survivable mapping és a részgráf összehúzás

1. A „survivable” tulajdonság gráfelméleti megfogalmazása:

Egy leképezés „survivable” az egyszeres él-hibákra $\Leftrightarrow$
$\neg\exists$ vágás melynek éleihez rendelt (fizikai rétegbeli) utak lefoghatóak egyetlen ponttal.

2. A „survivable” tulajdonsága a leképezésnek invariáns a kétszeresen él-összefüggő részgráf összehúzás műveletre.



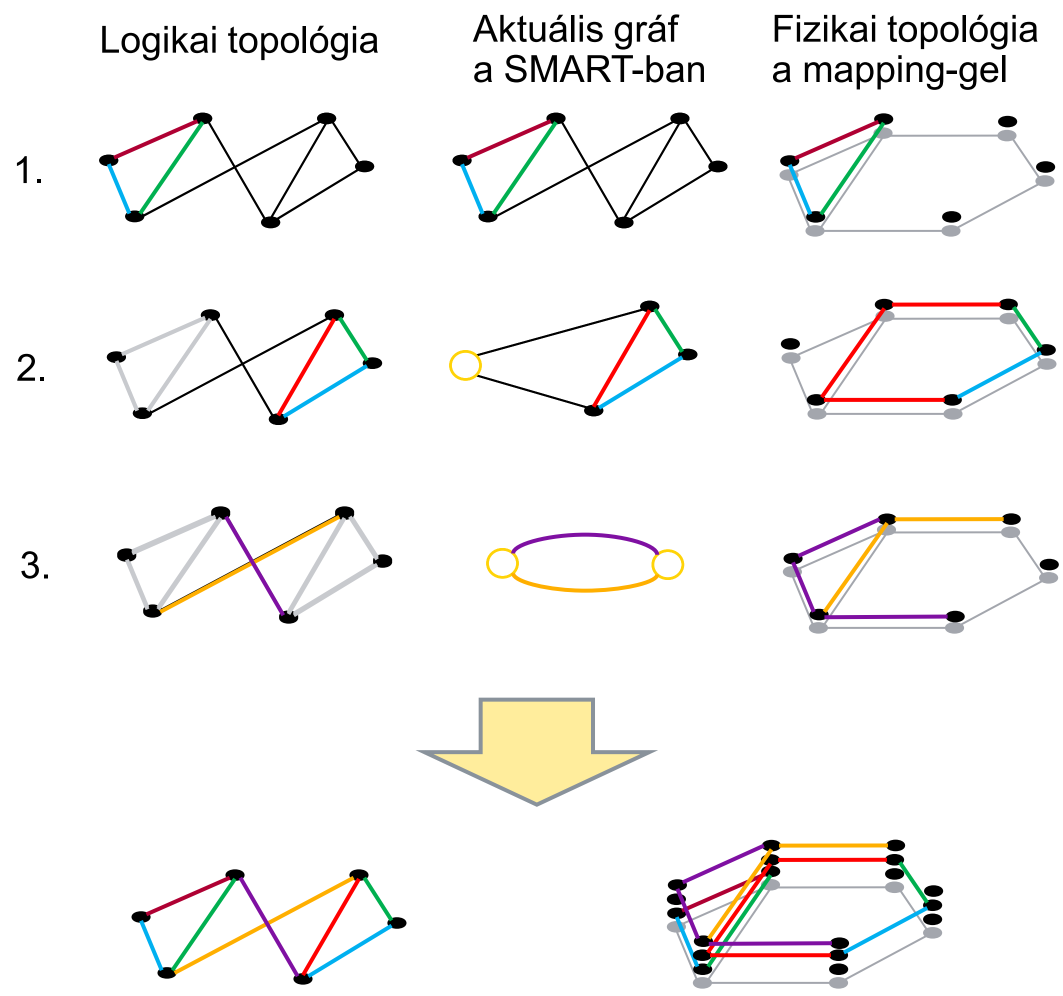
Gráf-elméleti megközelítés
SMART algoritmus

SMART algoritmus:

Step 1: Kiválasztunk a logikai topológiában egy 2-élösszefüggő részgráfot, és keresünk a benne lévő éleknek egy, a feltételeknek megfelelő leképezését a fizikai topológiára. (Amennyiben az él végpontja részgráf összehúzás során keletkezett, az élnek megfelelő utat az él eredeti, összehúzás előtti végpontjai között keressük.)
Ha nem létezik leképezhető részgráf, GOTO Step 3.

Step 2: Húzzuk össze a kiválasztott részgráfot egy ponttá, majd GOTO Step 1.

Step 3: Ha a gráfunk már csak egyetlen csúcsból áll, akkor az idáig vezető lépések során konstruált leképezések úniója egy megfelelő leképezését adja a teljes logikai topológiának. Ha nem, akkor a survivable mapping nem lehetséges.



Gráf-elméleti megközelítés
SMART algoritmus

M. Kurant, P. Thiran :

“Survivable Routing of Mesh Topologies in IP-over-WDM Networks by Recursive Graph Contraction”

IEEE J. Select. Areas Commun., Vol.25, No.5, pp.922-933, June 2007

A SMART algoritmus pszeudo-kódja:

Step 1: Kiindulásként a *G* gráf megegyezik a teljes logikai topológiával, $M_A=\emptyset$, $A=\emptyset$;

Step 2: Válasszunk egy 2-élösszefüggő részgráfot, $G_{sub} = (V_{sub}, B)$, és keressük az éleinek egy alkalmas leképezését (M_B). HA ilyen leképezés nem létezik, RETURN M_A és $G = G{\downarrow}A$; END;

Step 3: Adjuk hozzá a kapott leképezéseket a korábban konstruáltak M_A halmazához, azaz $M_A := M_A \cup M_B$;

Step 4: Hajtsuk végre a gráf összehúzást a választott részgráfra: $G := G{\downarrow}B$;

Step 5: HA *G* egyetlen csúcsból áll, RETURN M_a; END.

Step 6: GOTO Step 2

Sajnos nem polinom idejű:

A 2. lépés részgráf és leképezés keresése exponenciális futásidejű, így nagyobb hálózatokra nem alkalmazható.

→ alkalmas polinom idejű heurisztikára van szükség.

Gráf-elméleti megközelítés
SMART-H algoritmus

“Survivable Mapping Algorithm by Ring Trimming”

A SMART algoritmus pszeudo-kódja:

Step 1: Kiindulásként a G gráf megegyezik a teljes logikai topológiával, $M_A=\emptyset$, $A=\emptyset$;

Step 2: Válasszunk egy 2-élösszefüggő részgráfot, $G_{sub} = (V_{sub}, B)$, és keressük az éleinek egy alkalmas leképezését (M_B). HA ilyen leképezés nem létezik, RETURN M_A és $G = G\downarrow A$; END;

Step 3: Adjuk hozzá a kapott leképezéseket a korábban konstruáltak M_A halmazához, azaz $M_A := M_A \cup M_B$;

Step 4: Hajtsuk végre a gráf összehúzást a választott részgráfra: $G := G\downarrow B$;

Step 5: HA G egyetlen csúcsból áll, RETURN M_a; END.

Step 6: GOTO Step 2

Step 2 variáció:

A G_{sub} gráfként a gráf egy körét keressük, majd a kör éleit a következők szerint képezzük le:

1. A fizikai gráf éleihez súlyokat rendelünk, kezdetben minden él súlya 1.
2. Minden iterációs lépésben a az éleket a végpontok közti legrövidebb (legkisebb súlyú) útra képezzük le.
3. Ha minden fizikai élt csak egyszer használtunk a kör éleinek leképezésénél, akkor a leképezés megfelelő (survivable).
4. Ha van fizikai él, amit több út is használ, annak az élnek a súlyát növeljük meg, majd végezzük el ismét a legrövidebb út kereséseket.
5. Ha a sikertelen útkeresések száma túllép egy előre meghatározott határt, a leképezés keresését sikertelennek nyilvánítjuk.

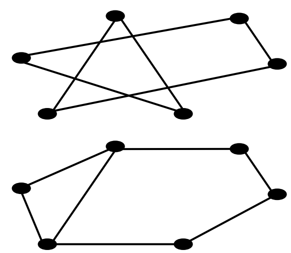
+

Step 7 (opcionális): Ha az összehúzott gráf csúcsainak száma egy előre meghatározott határ alá esik, kereshetjük a leképezést az összes lehetséges leképezés próbálgatásával is.

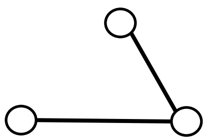
Gráf-elméleti megközelítés
A gráfelméleti heurisztika előnyei

- Ha nem is találja meg a teljes logikai topológia leképezését, a kapott parciális megoldás is használható.
- Ha sikertelen is a leképezési kísérlet, a kapott (rész-) eredmény segítséget nyújt a szükséges bővítések (logikai élek hozzáadása) helyének meghatározásához.
- Többszörös él-hibák esetére is általánosítható.

Nem létezik jó leképezés



Nem 2-élösszefüggő
a logikai topológia



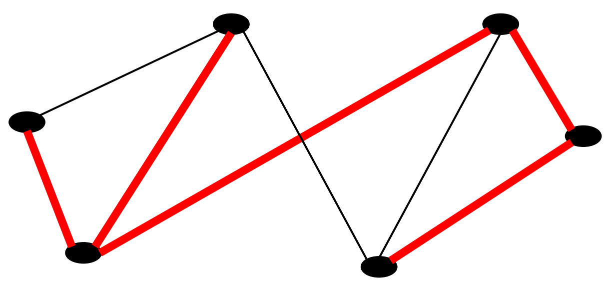
Gráf-elméleti megközelítés
CIRCUIT-SMART, CUTSET-SMART

K. Thulasiraman, M. Javed, G. Xue :

“Circuits/Cutsets Duality and a Unified Algorithm Framework for Survivable Logical Topology Design in IP-over-WDM Optical Networks”

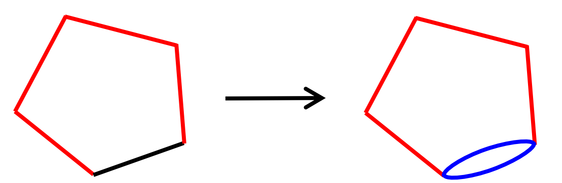
in Proc. IEEE INFOCOM, 2009, pp.1026-1034

Fundamentális körökre alkalmazzuk a leképezés-összehúzás operációt



Mapping lépés:

1. Leképezzük éldiszjunkt módon a kiválasztott kör maximális számú élét.
2. Ha a kör minden élét le tudtuk képezni, kész vagyunk.
3. Ha nem, a maradék élek mellé hozzáadunk egy új párhuzamos logikai élt, és a két élt éldiszjunkt módon leképezzük.



Gráf-elméleti megközelítés
Többszörös él-hibák esete

M. Kurant, P. Thiran :
"Survivable Routing in IP-over-WDM Networks in the Presence of Multiple Failures"
http://arxiv.org/abs/cs.NI/0604053, 2006

A SMART algoritmus pszeudo-kódja:

Step 1: Kiindulásként a G gráf megegyezik a teljes logikai topológiával, $M_A=\emptyset$, $A=\emptyset$;

Step 2: Válasszunk egy 2-élösszefüggő részgráfot, $G_{sub} = (V_{sub}, B)$, és keressük az éleinek egy alkalmas leképezését (M_B). HA ilyen leképezés nem létezik, RETURN M_A és $G = G{\downarrow}A$; END;

Step 3: Adjuk hozzá a kapott leképezéseket a korábban konstruáltak M_A halmazához, azaz $M_A := M_A \cup M_B$;

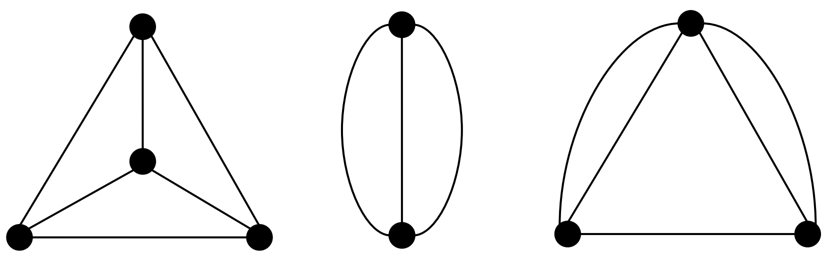
Step 4: Hajtsuk végre a gráf összehúzást a választott részgráfra: $G := G{\downarrow}B$;

Step 5: HA G egyetlen csúcsból áll, RETURN M_a; END.

Step 6: GOTO Step 2

Általánosítható *k*-élhibára:

k él-hibára való védelem esetén a topológiákra vonatkozó feltétel (*k*+*1*)-élösszefüggőségre változik. Ennek megfelelően az algoritmusban is a (*k*+*1*)-élösszefüggő részgráfokra kell alkalmazni a diszjunkt útkeresést és gráf-összehúzást.



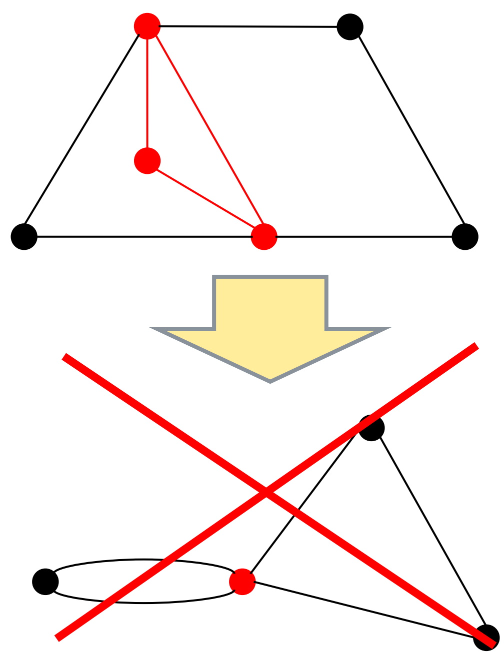
Gráf-elméleti megközelítés
Egyszeres csúcs-hibák esete

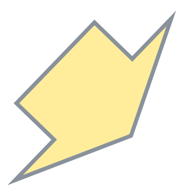
L. Zhang, N. Zhang :

“Survivable virtual topology design for single-node failure”

Proc. IEEE ICCRD (2009), pp. 465-467.

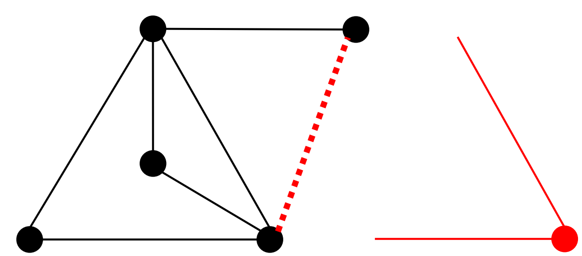
Az eddigi leképezés-összehúzás módszer nem működik csúcs-hibák esetén, mivel a (csúcs-)összefüggőség nem invariáns az összehúzás műveletre.





Leképezés-csúcselhagyás módszer:

1. Kiválasztunk egy csúcsot, és a rá illeszkedő éleket csúcs-független utakra képezzük le.
2. A csúcsot elhagyjuk a gráfból.
3. Ha a 2-összefüggőség megtartásához szükséges, az elhagyott csúcs szomszédjait összekötjük új éllel, mely él örökli az elhagyott csúcson keresztüli összeköttetés leképezését.



Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete

A két csúcs-hibát kivédő survivable mapping alapfeltétele, hogy mind a logikai, mind a fizikai topológia 3-összefüggő legyen.

A 3-összefüggő gráfoknak azonban van néhány jó tulajdonsága, amit kihasználhatunk alkalmas heurisztika konstruálásához.

Tétel 1:

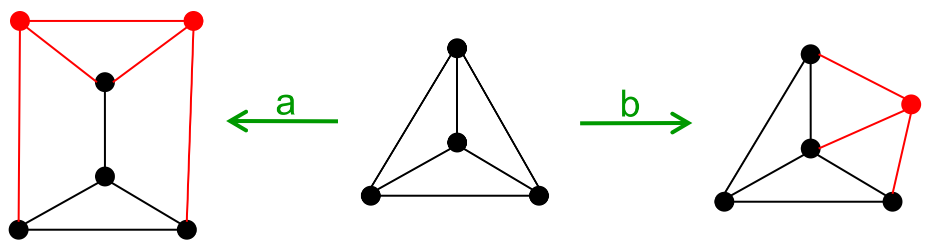
Egy *G* gráf 3-összefüggő akkor és csak akkor, ha *G* felépíthető a K_4 gráfból az alábbi két művelet alkalmazásával, illetve élek hozzáadásával:

(a) Fogunk két nem párhuzamos élt, ketté osztjuk őket egy-egy csúccsal, majd a két új csúcsot összekötjük egy éllel;

(b) Fogunk egy élt, ketté osztjuk egy csúccsal, majd az új csúcsot összekötjük egy, a felosztott él végpontjaitól különböző csúccsal.

Tétel 2 (Chartrand et al.):

Minden 3-összefüggő gráfban, melynek minden csúcsa legalább negyed fokú, létezik legalább egy csúcs, melynek elhagyásával a gráf 3-összefüggő marad.



Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete

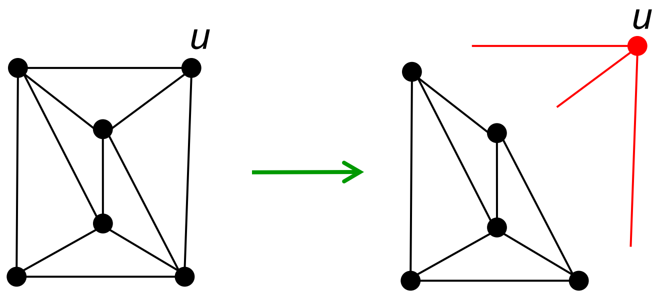
3 féle gráf redukciós lépést konstruálunk, melyek tartalmazzák az elhagyott élek leképezésének szükséges feltételeit is:

A) Csúcs elhagyása

B) Egy csúcs leemelése (a Tétel 1-ben szereplő (b) művelet megfordítása)

C) Két csúcs leemelése (a Tétel 1–ben szereplő (a) művelet megfordítása)

A)



Az élek leképezése:

I. A csúcshoz illeszkedő élek közül 3-at leképezünk csúcs-független utakra, a maradék éleket (ha vannak) pedig tetszőleges utakra.

II. A megmaradt 3-összefüggő gráfot leképezzük survivable módon (rekurzívan alkalmazva a három lépést).

III. A két leképezés uniója a teljes topológia egy jó leképezését adja.

Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete

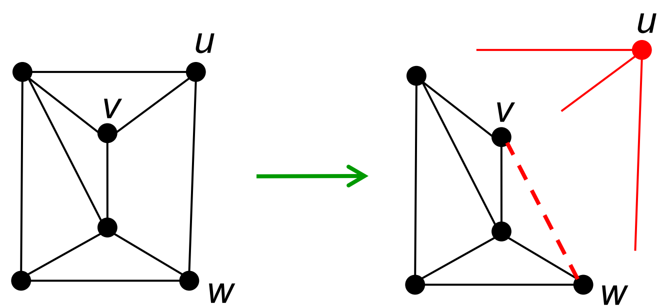
3 féle gráf redukciós lépést konstruálunk, melyek tartalmazzák az elhagyott élek leképezésének szükséges feltételeit is:

A) Csúcs elhagyása

B) Egy csúcs leemelése (a Tétel 1-ben szereplő (b) művelet megfordítása)

C) Két csúcs leemelése (a Tétel 1–ben szereplő (a) művelet megfordítása)

B)



Az élek leképezése:

I. A csúcshoz illeszkedő élek közül 3-at leképezünk csúcs-független utakra, a maradék éleket (ha vannak) pedig tetszőleges utakra.

II. A csúcs elhagyását követően megmaradó gráfba „virtuális élt” illesztünk, hogy megőrizzük a 3-összefüggőséget.

III. A megmaradt 3-összefüggő gráfot leképezzük survivable módon (rekurzívan alkalmazva a három lépést), ahol a virtuális él leképezését adottnak tekintjük (az elhagyott csúcs éleinek leképezéséből örökölt út).

IV. A két leképezés uniója a teljes topológia egy jó leképezését adja.

Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete

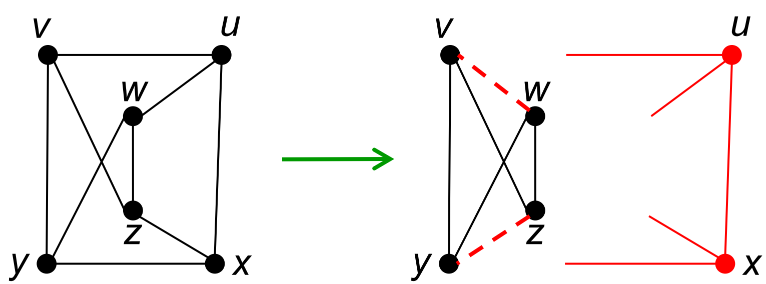
3 féle gráf redukciós lépést konstruálunk, melyek tartalmazzák az elhagyott élek leképezésének szükséges feltételeit is:

A) Csúcs elhagyása

B) Egy csúcs leemelése (a Tétel 1-ben szereplő (b) művelet megfordítása)

C) Két csúcs leemelése (a Tétel 1–ben szereplő (a) művelet megfordítása)

C)



Az élek leképezése:

I. A két elhagyandó csúcs 5 élét a következő feltételekkel képezzük le:
   a) Az *u* csúcs éleihez *(uv, uw* és *ux*) rendelt utak legyenek csúcs-diszjunktak.
   b) Az *x* csúcs éleihez *(xz, xz* és *ux*) rendelt utak legyenek csúcs-diszjunktak.
   c) Ne létezzen 2 csúcs, amelyek az *uv*, *uw*, *xy* és *xz* élekhez tartozó utak mindegyikét lefogják.

II. A csúcs elhagyását követően megmaradó gráfba illesszünk „virtuális éleket”, hogy megőrizzük a 3-összefüggőséget.

III. A megmaradt 3-összefüggő gráfot leképezzük survivable módon (rekurzívan alkalmazva a három lépést), ahol a virtuális élek leképezését adottnak tekintjük.

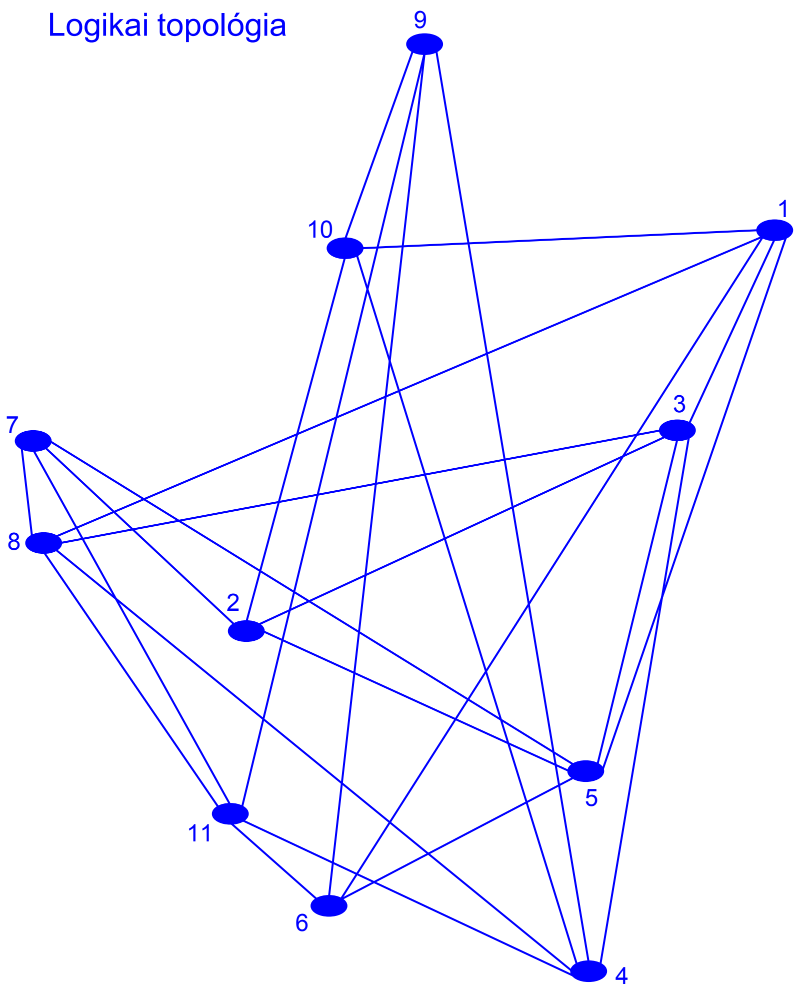
IV. A két leképezés uniója a teljes topológia egy jó leképezését adja.

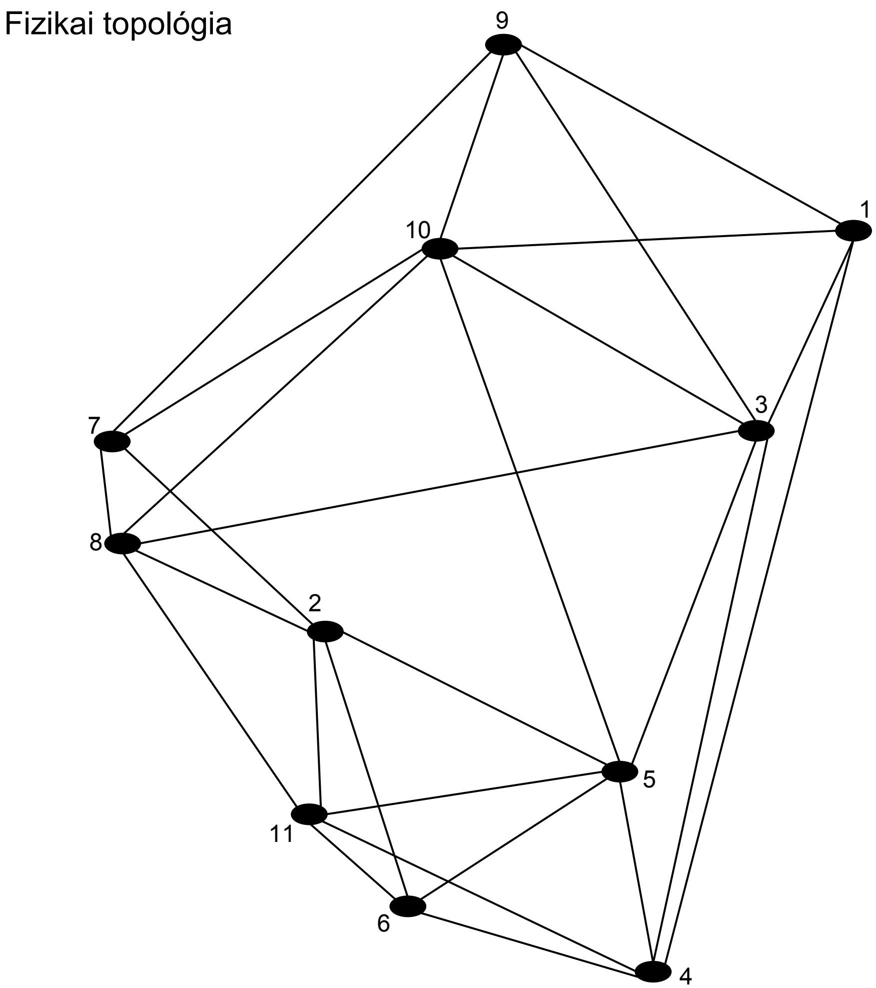
Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete

Az algoritmus vázlata:

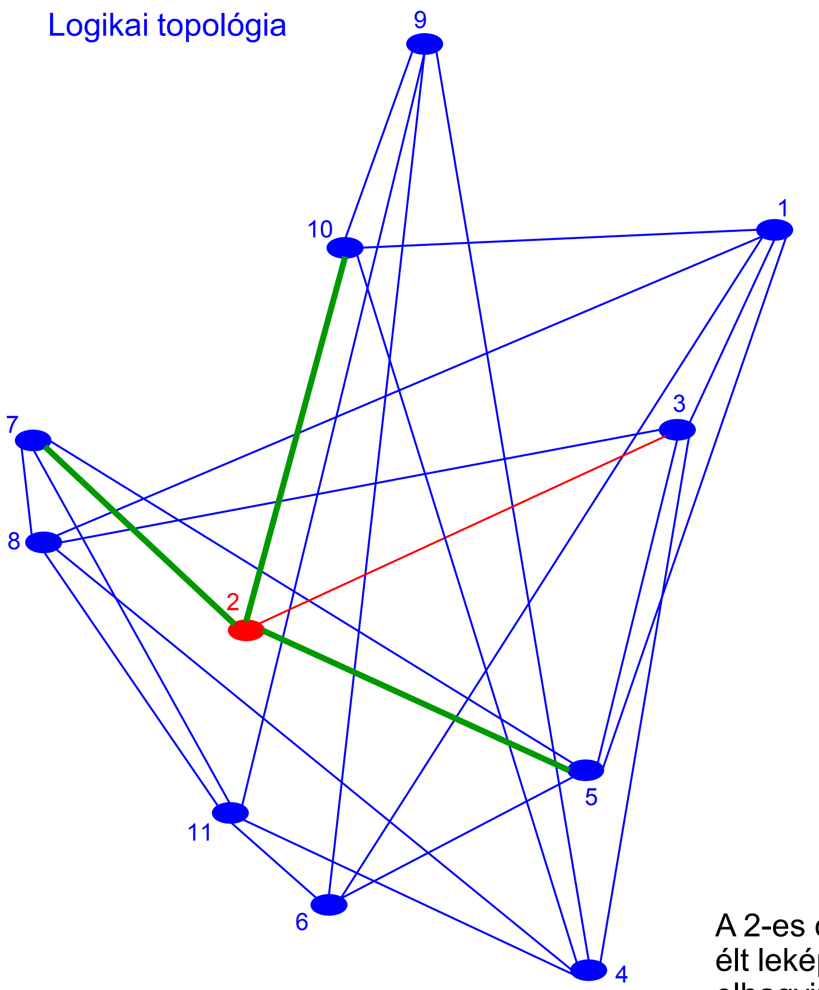
1. Amíg találunk olyan csúcsot, amely elhagyásával a 3-összefüggőség megmarad, alkalmazzuk rájuk a A típusú redukciós lépést.
2. Ha elhagyható csúcsot nem találtunk, alkalmazzuk a B típusú redukciós lépést a leemelhető csúcsokra, amíg találunk ilyet.
3. Ha a fenti két lépés nem alkalmazható, akkor a leemelhető csúcs-párokra alkalmazzuk a C típusú lépést.
4. Ha a maradék gráf még több, mint 4 csúcsú, de már egyik redukciós lépés sem alkalmazható, akkor eredménytelen volt a konstrukció.
5. Ha a maradék gráf izomorf a K_4 gráffal, akkor keressük meg a K_4 egy jó leképezését. Ha létezik, akkor kész vagyunk. Ha nem, akkor eredménytelen volt a konstrukció.

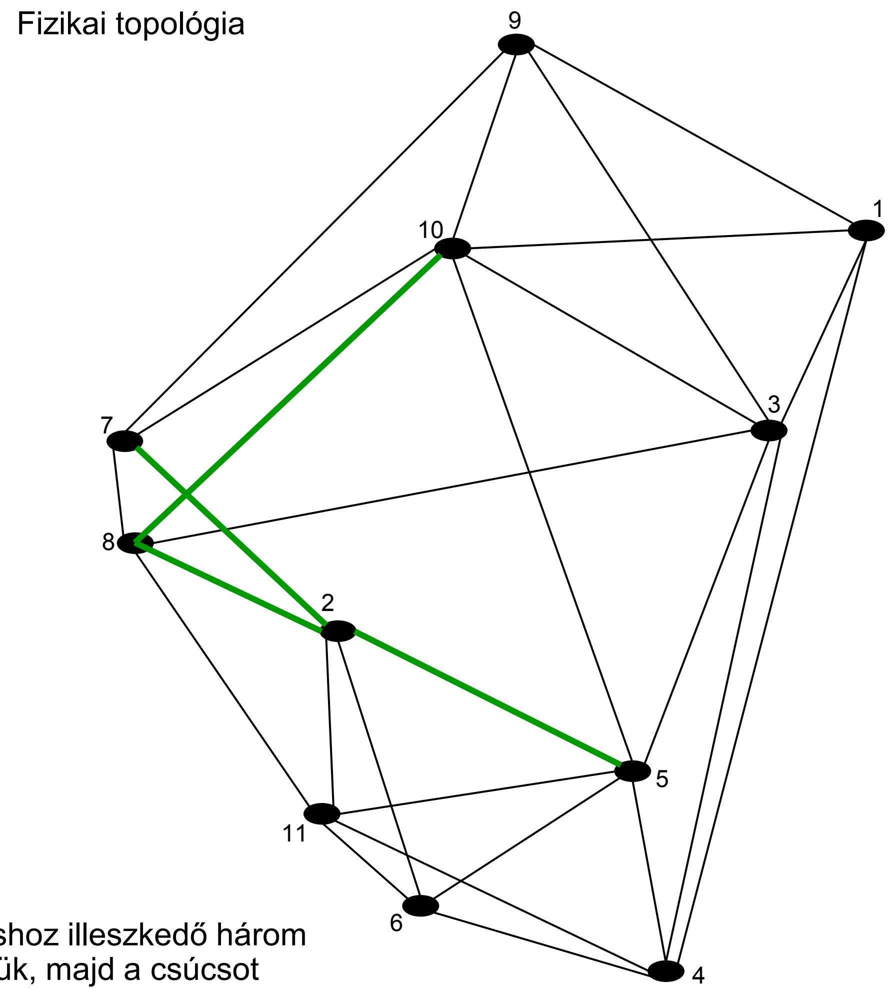
Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására





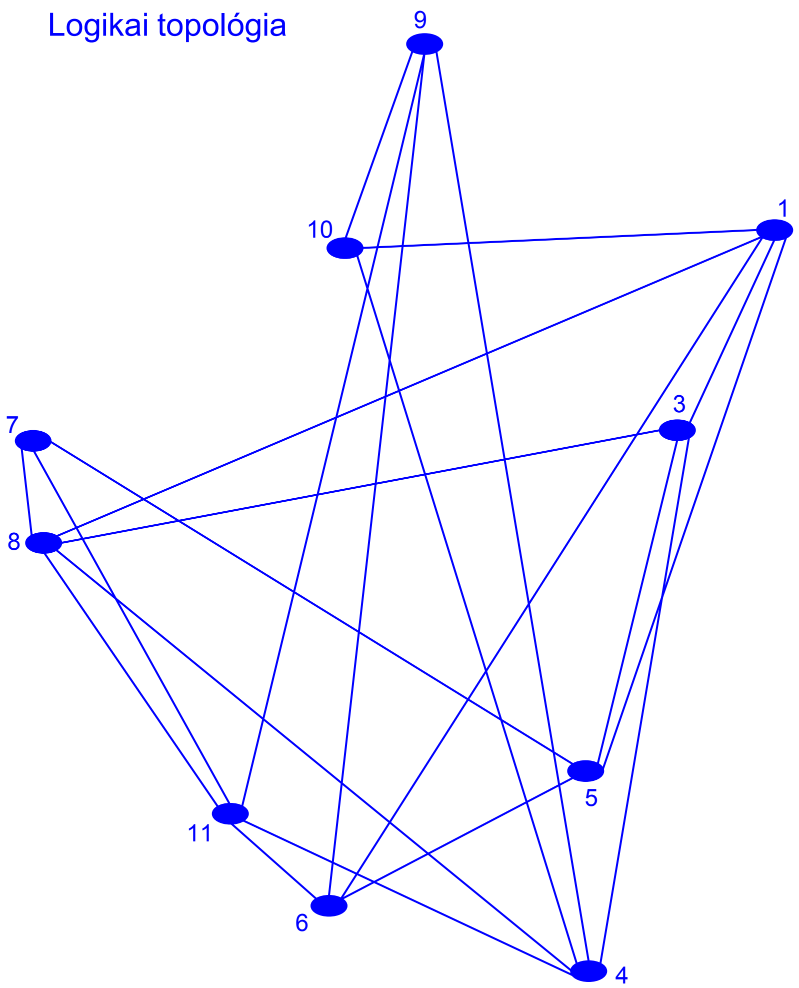
Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

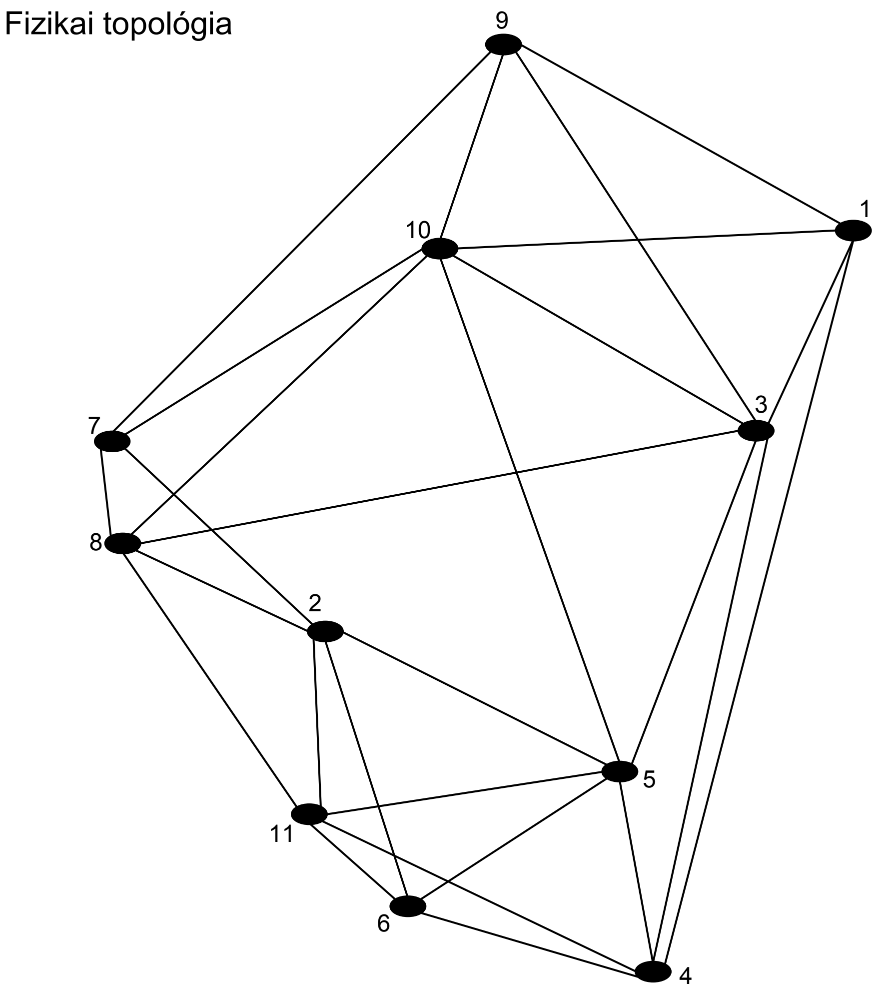




A 2-es csúcshoz illeszkedő három élt leképezzük, majd a csúcsot elhagyjuk. (A negyedik csúcs tetszőlegesen leképezhető.)

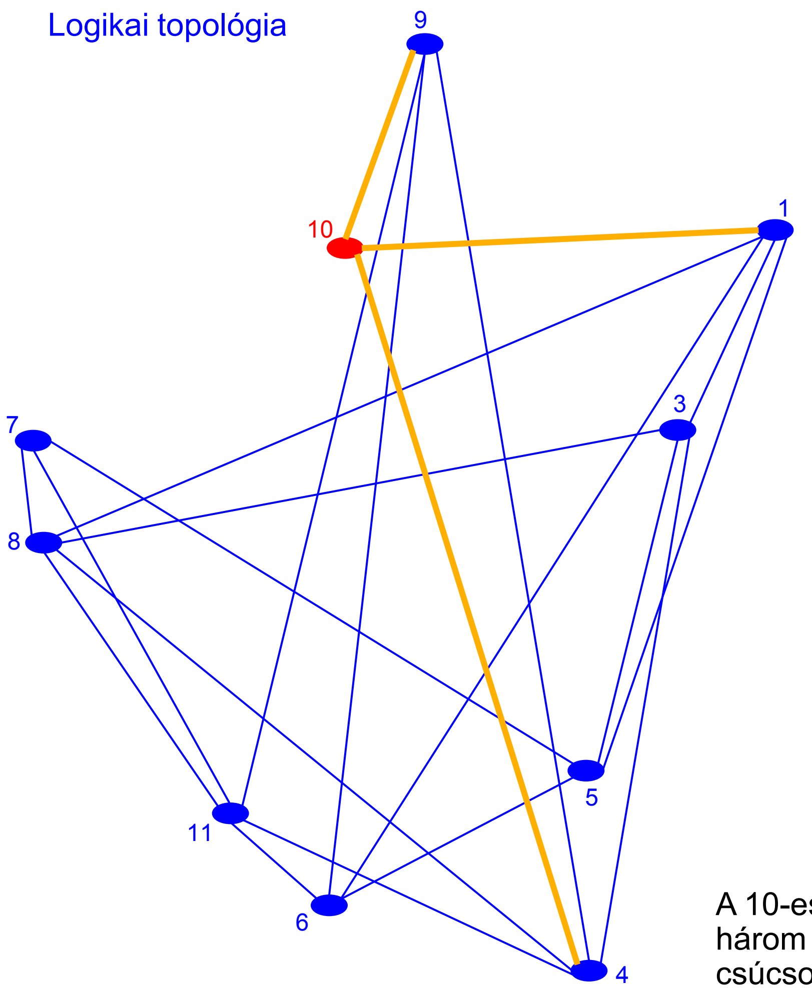
Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

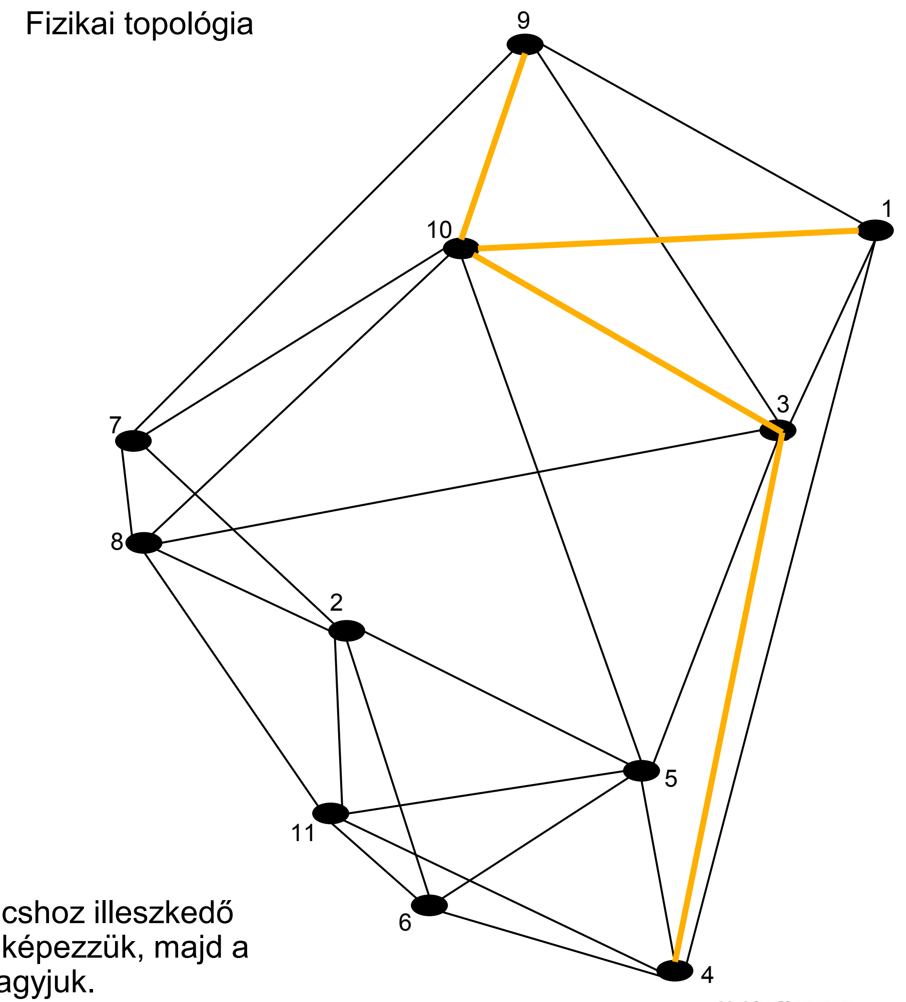




Gráf-elméleti megközelítés

Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására



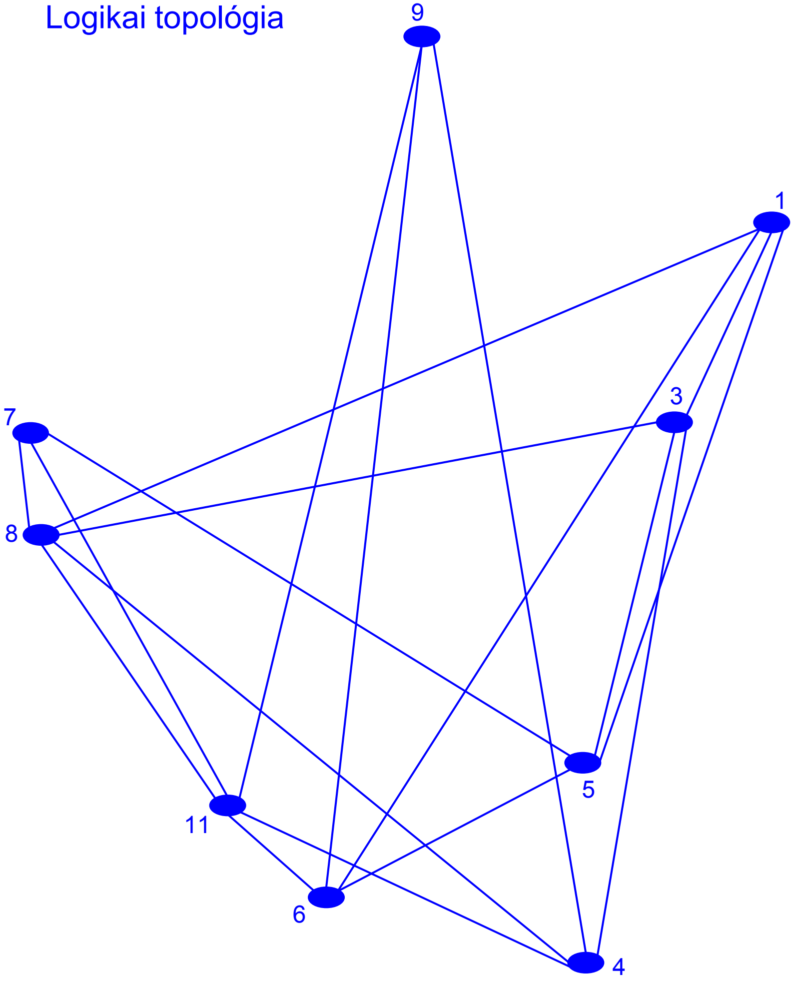


A 10-es csúcshoz illeszkedő három élt leképezzük, majd a csúcsot elhagyjuk.

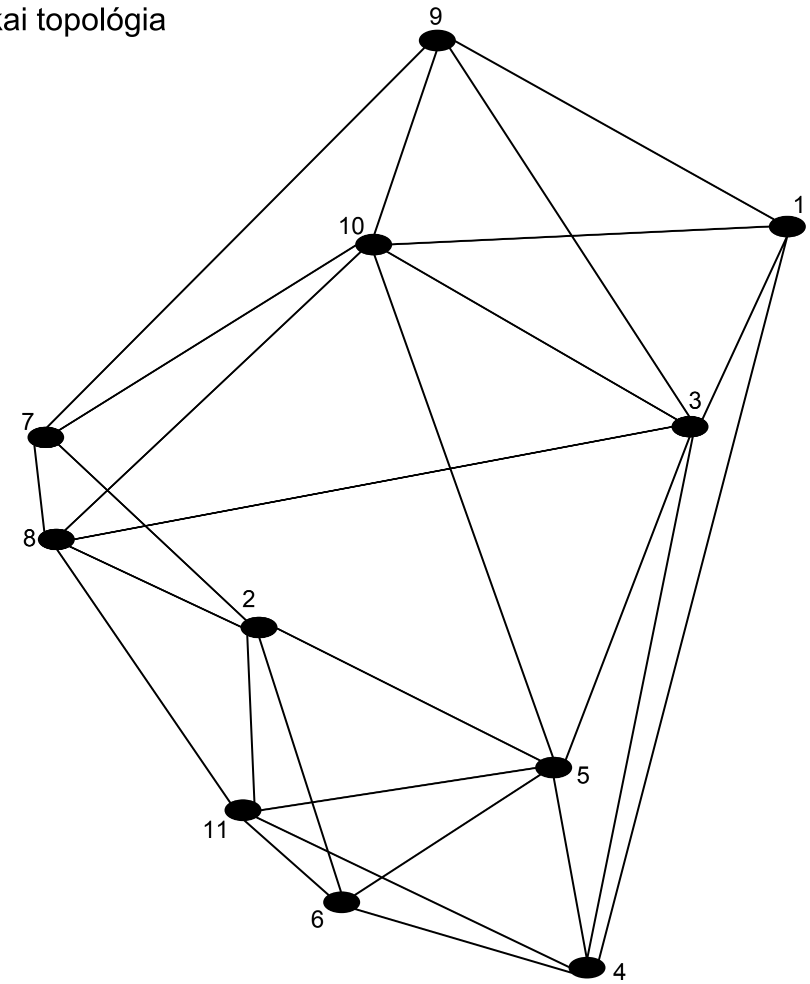
Gráf-elméleti megközelítés

Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

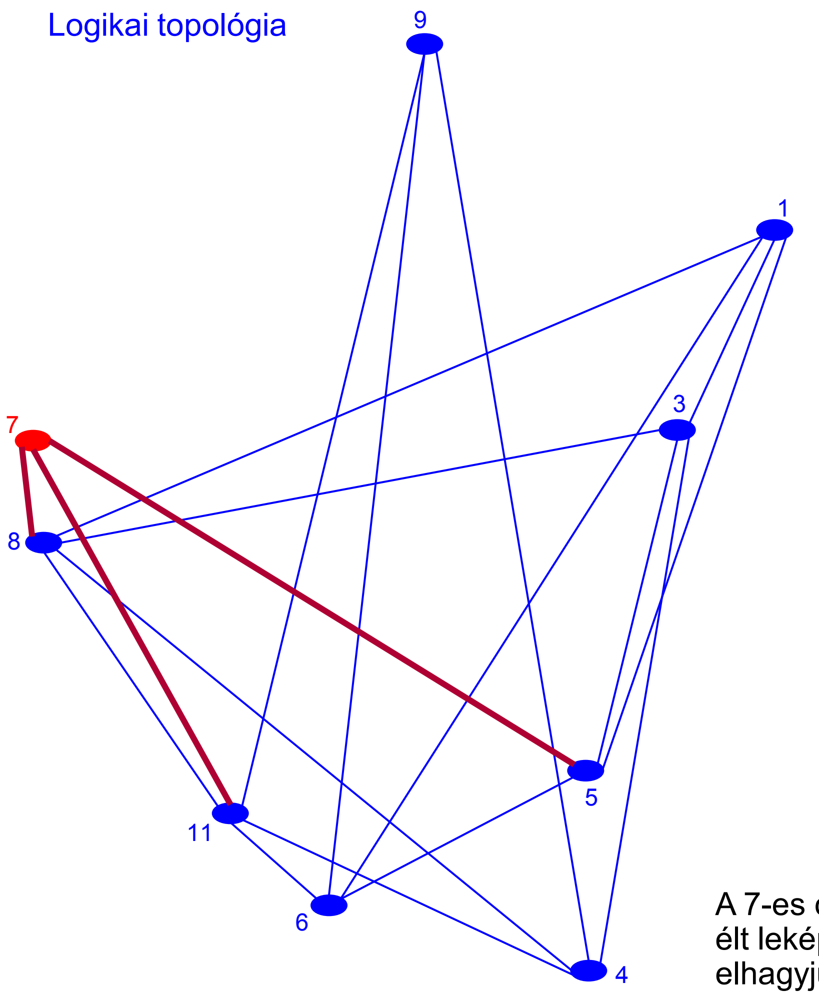
Logikai topológia

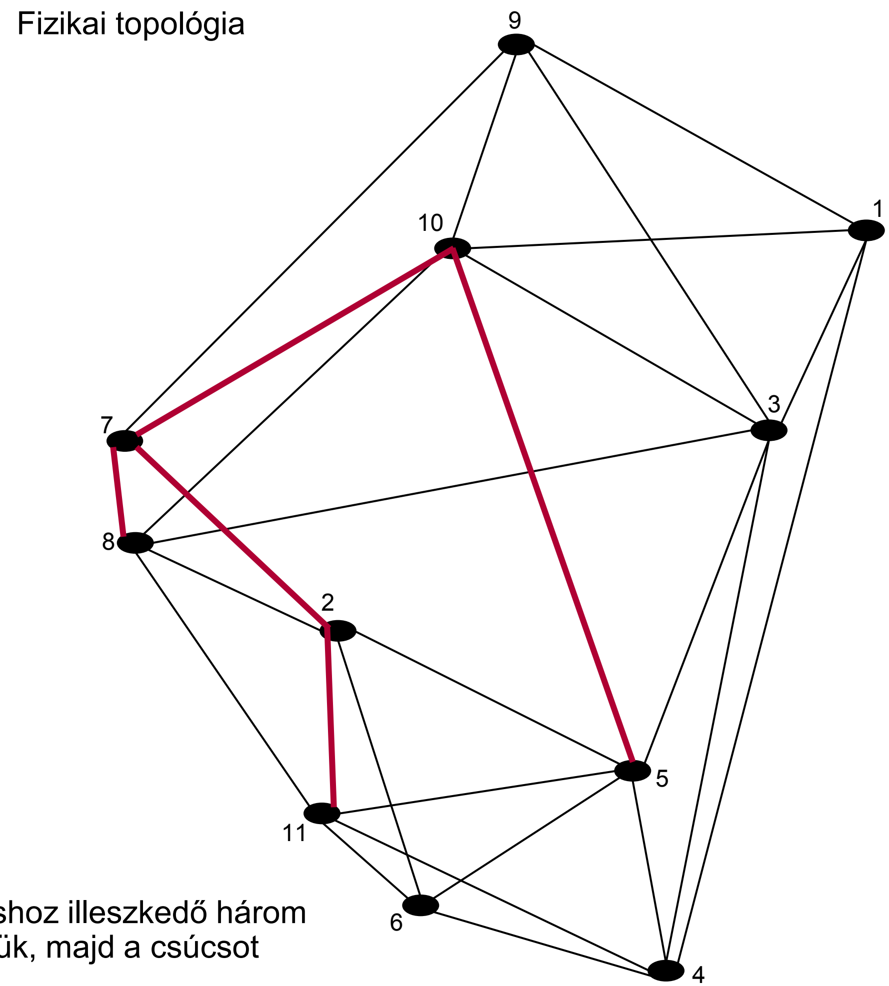


Fizikai topológia



Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

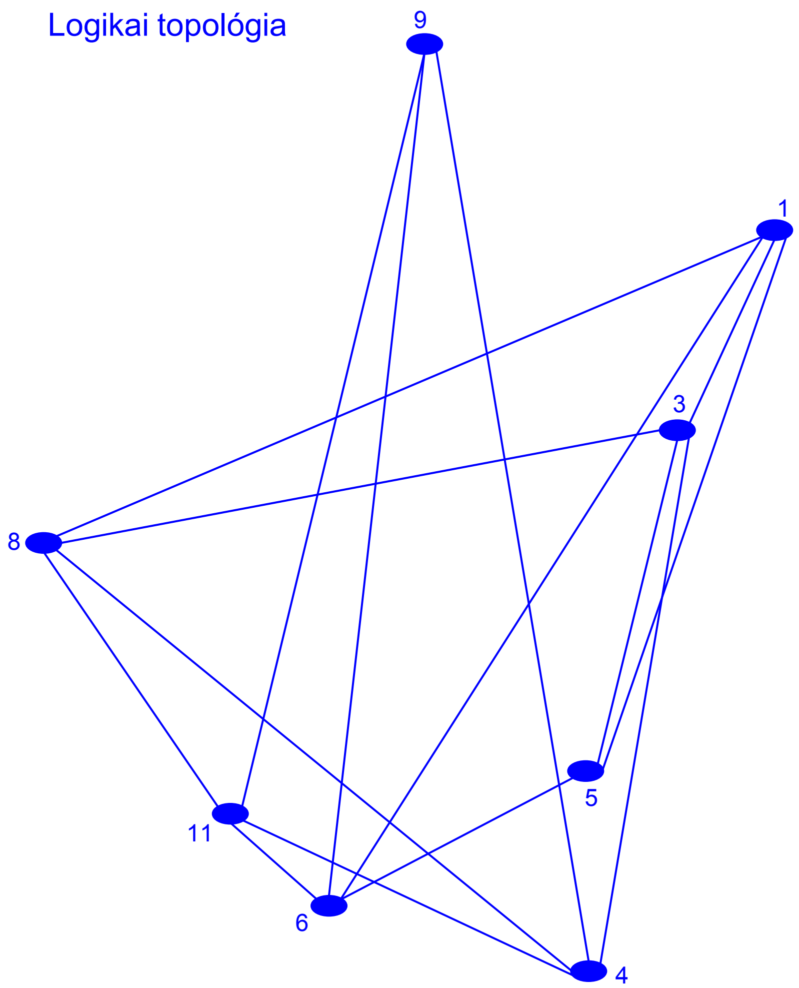




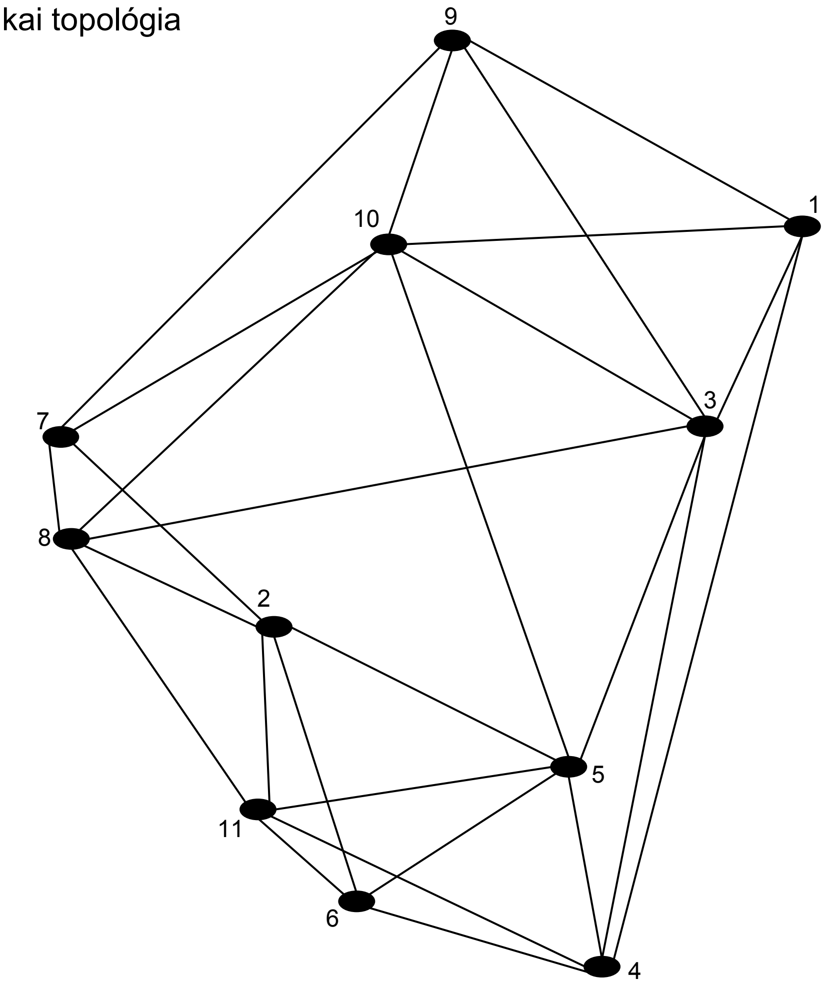
A 7-es csúcshoz illeszkedő három élt leképezzük, majd a csúcsot elhagyjuk.

Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

Logikai topológia



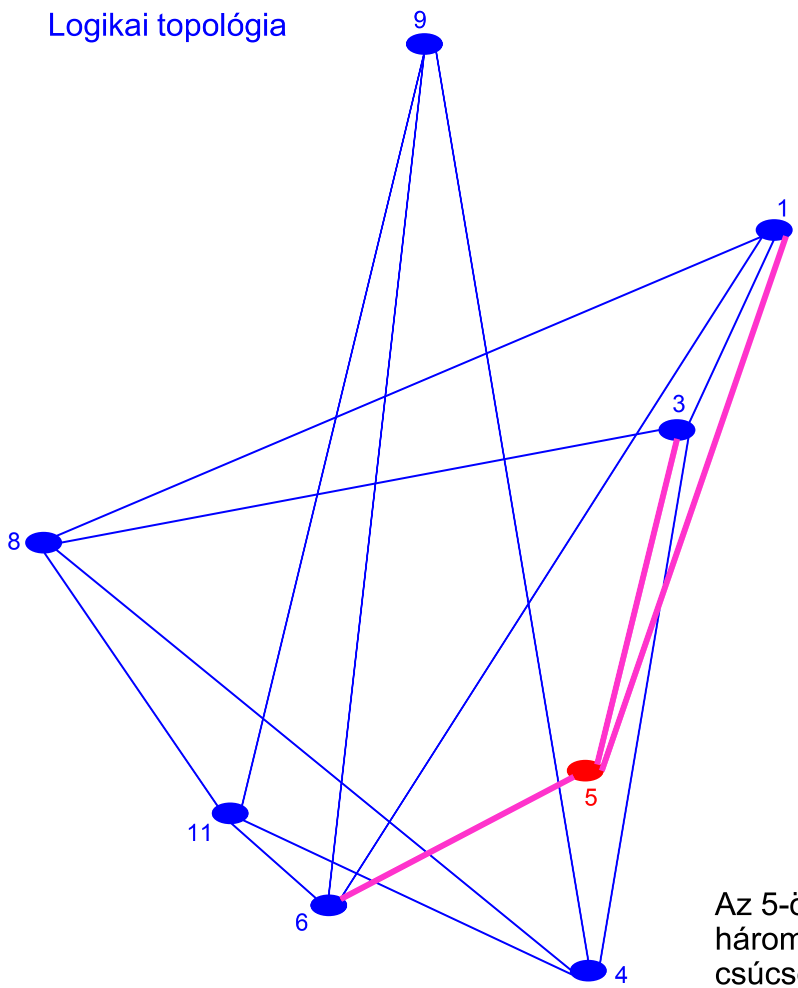
Fizikai topológia



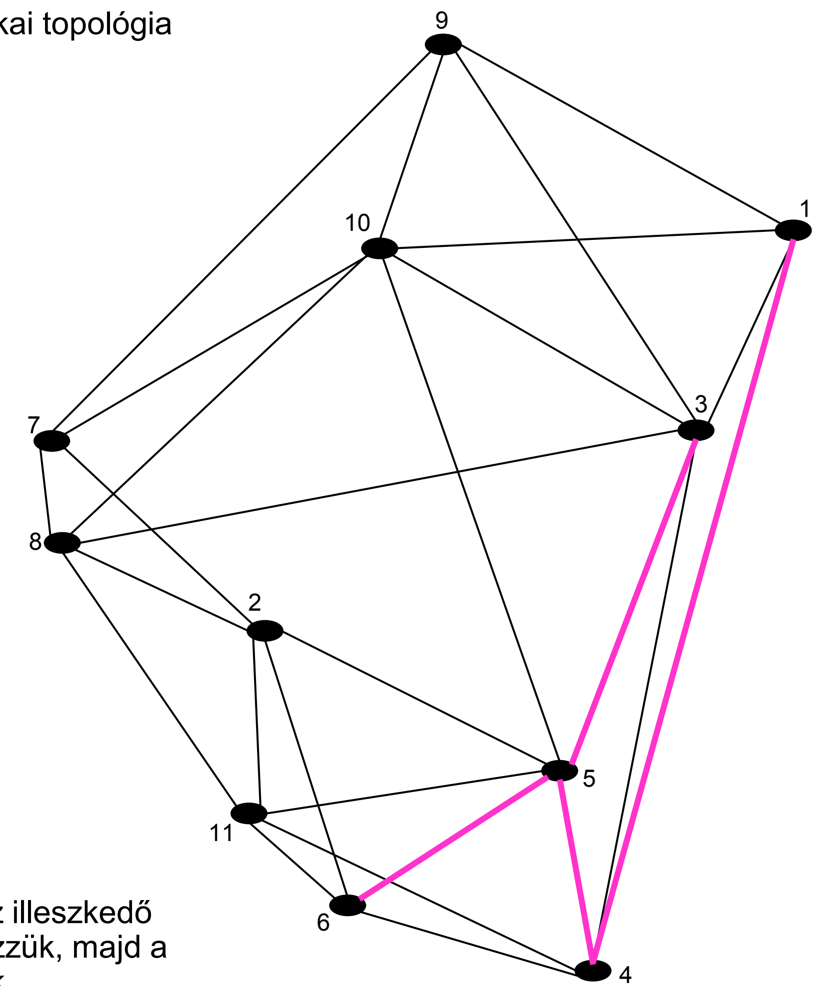
Gráf-elméleti megközelítés

Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

Logikai topológia

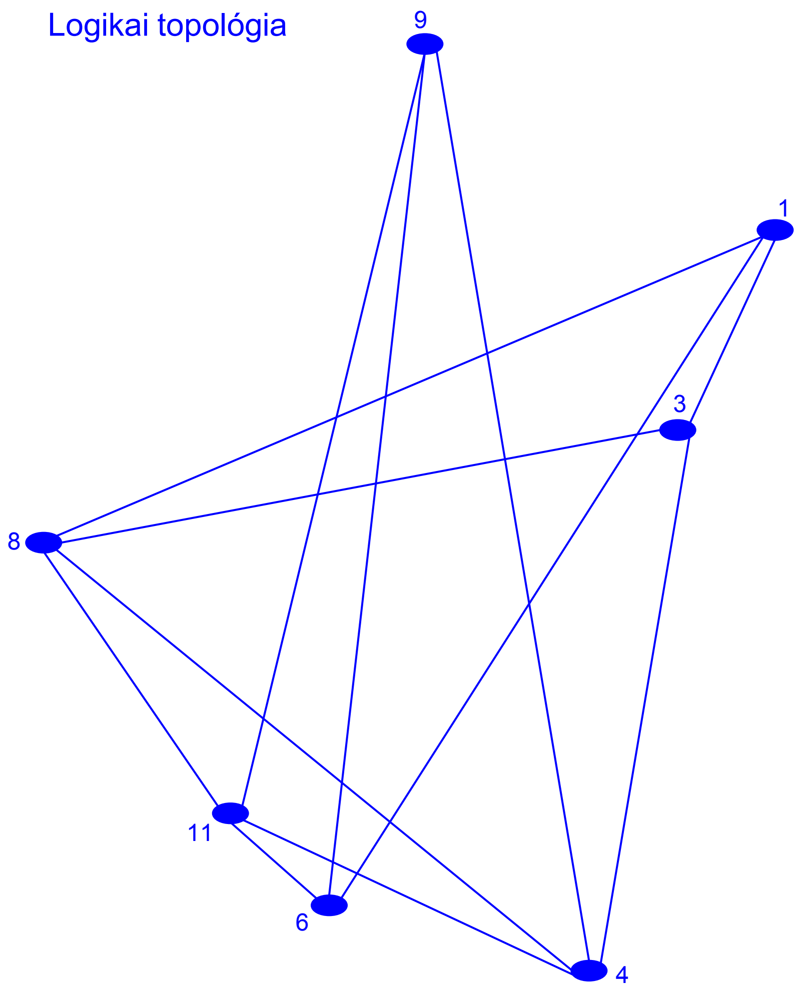


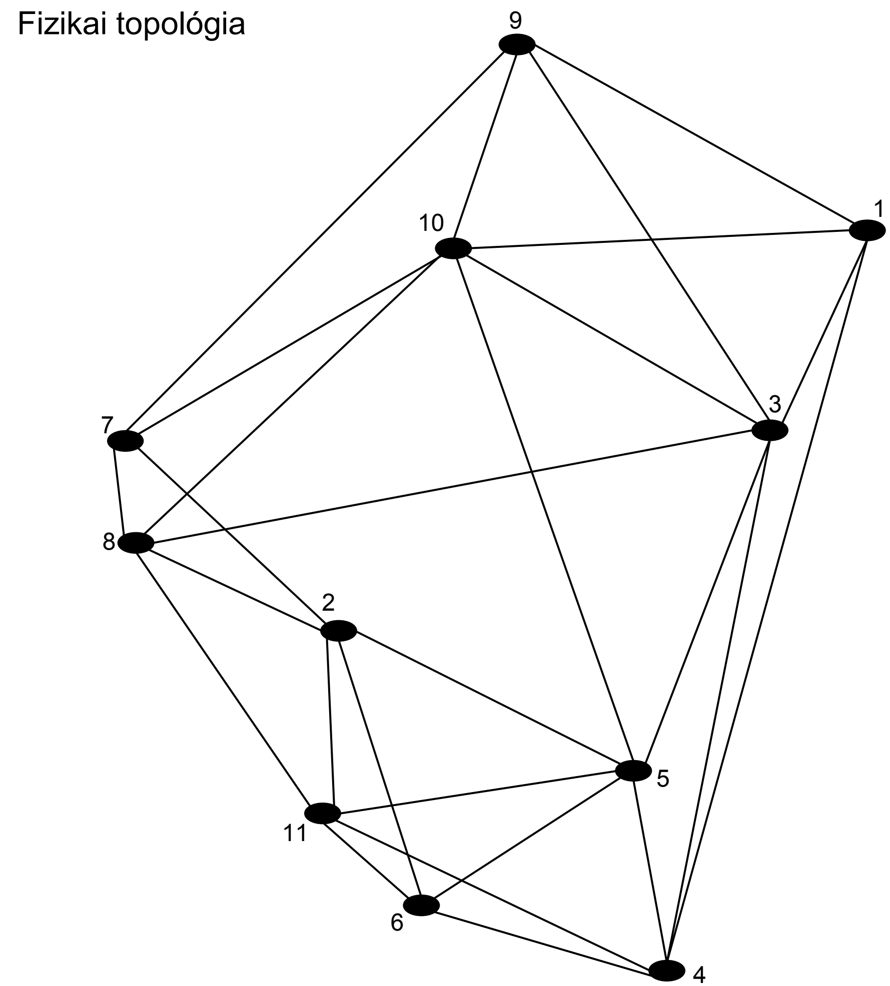
Fizikai topológia



Az 5-ös csúcshoz illeszkedő három élt leképezzük, majd a csúcsot elhagyjuk.

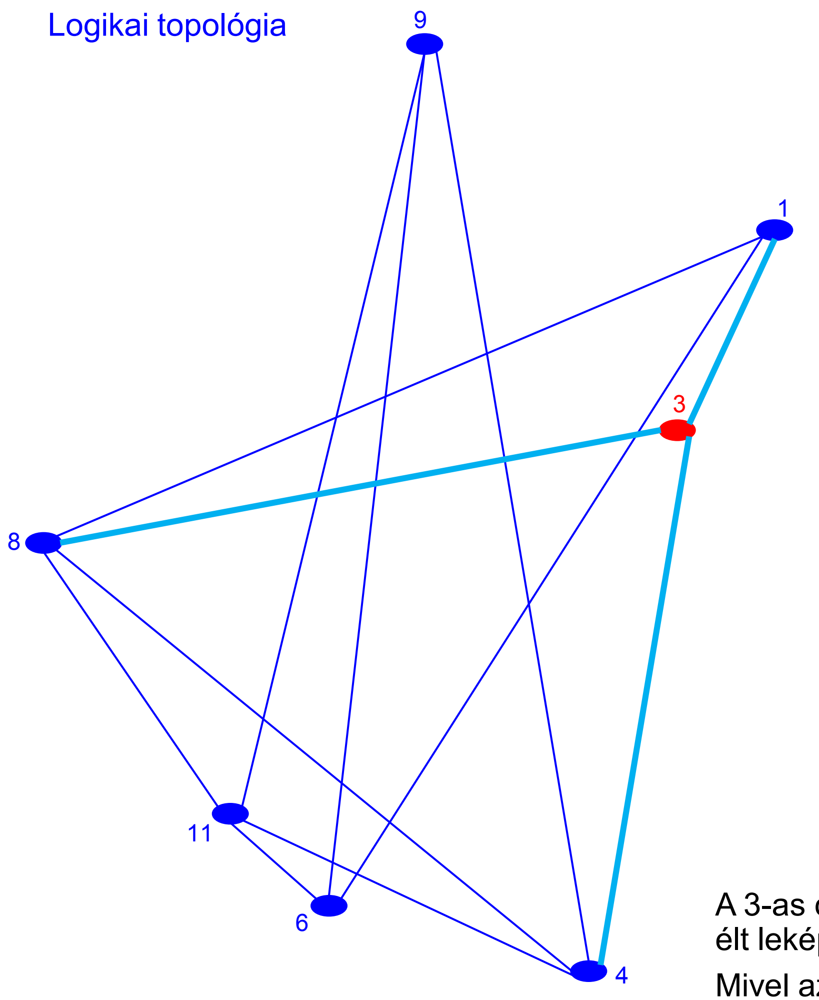
Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

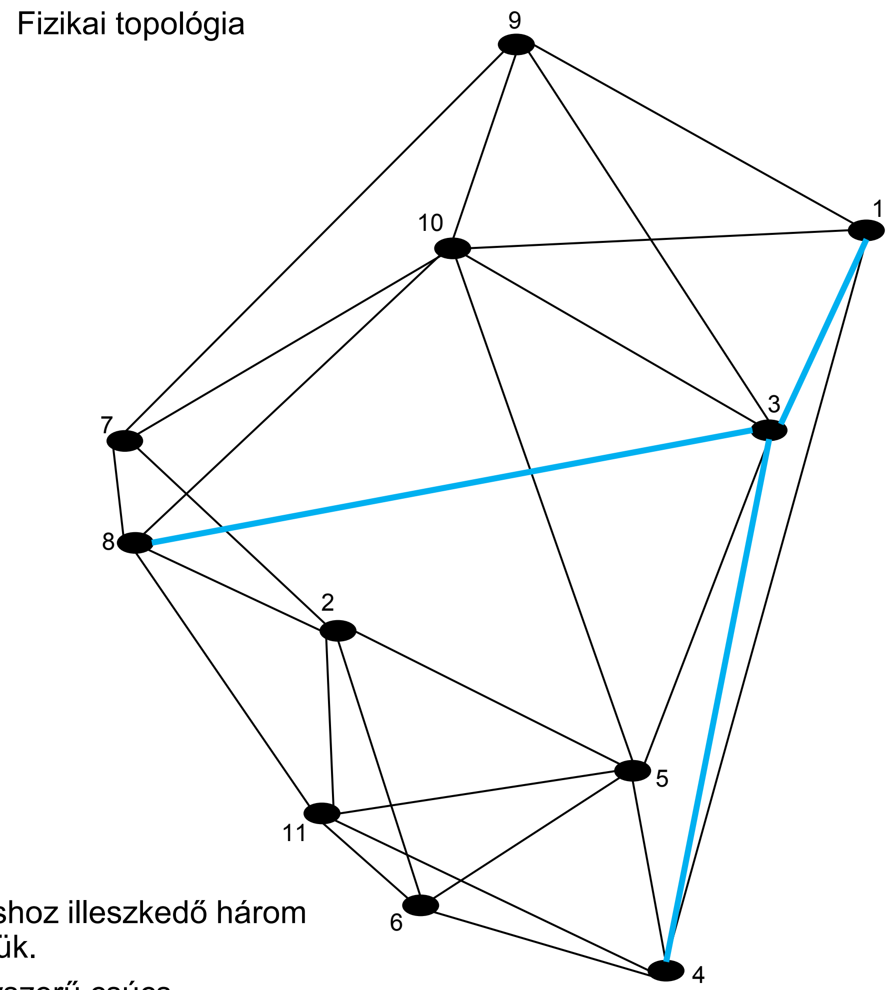




Gráf-elméleti megközelítés

Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

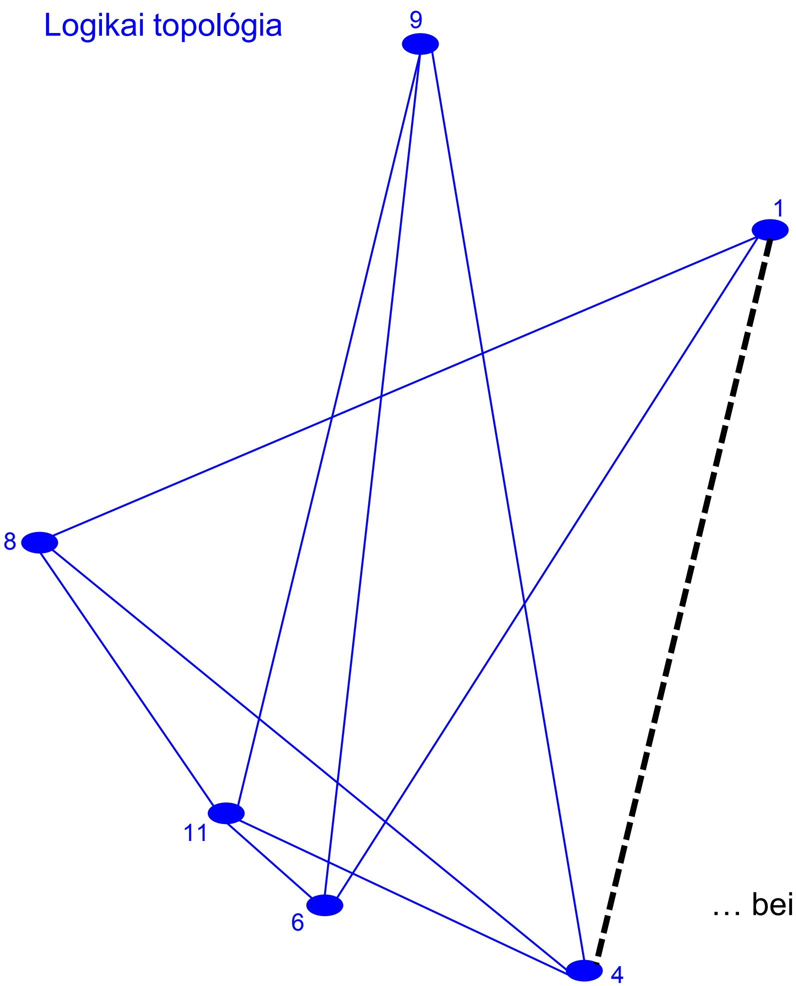


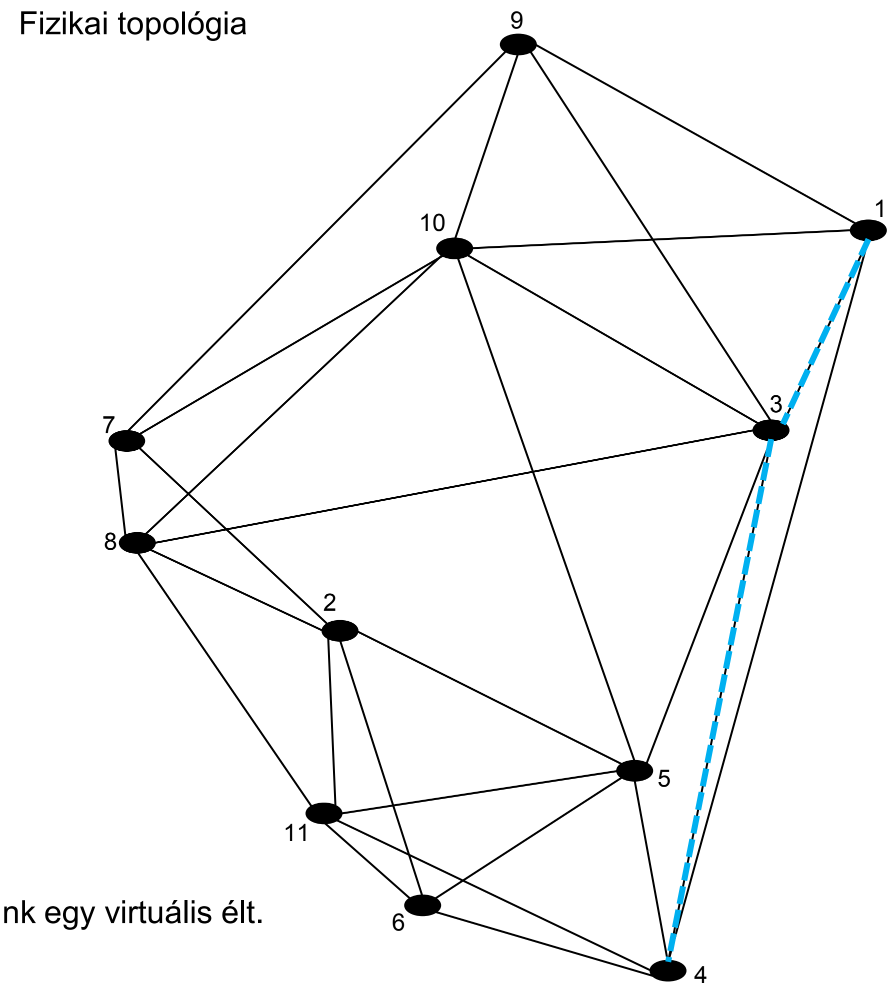


A 3-as csúcshoz illeszkedő három élt leképezzük.

Mivel az egyszerű csúcs-elhagyás nem alkalmazható …

Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

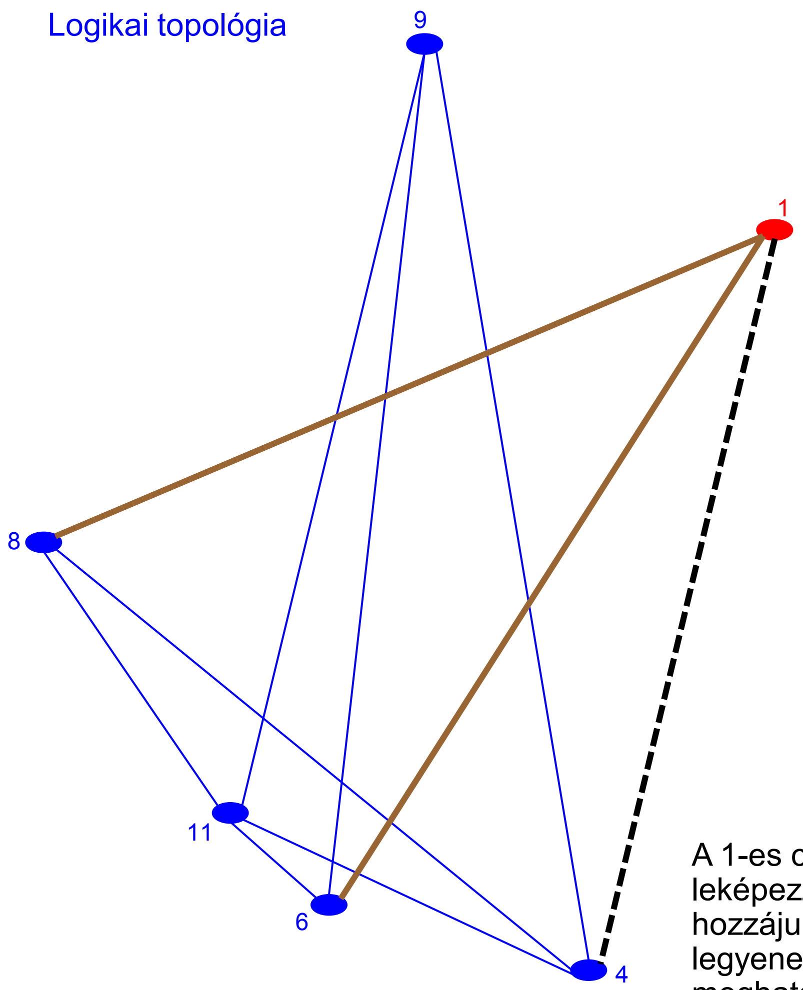


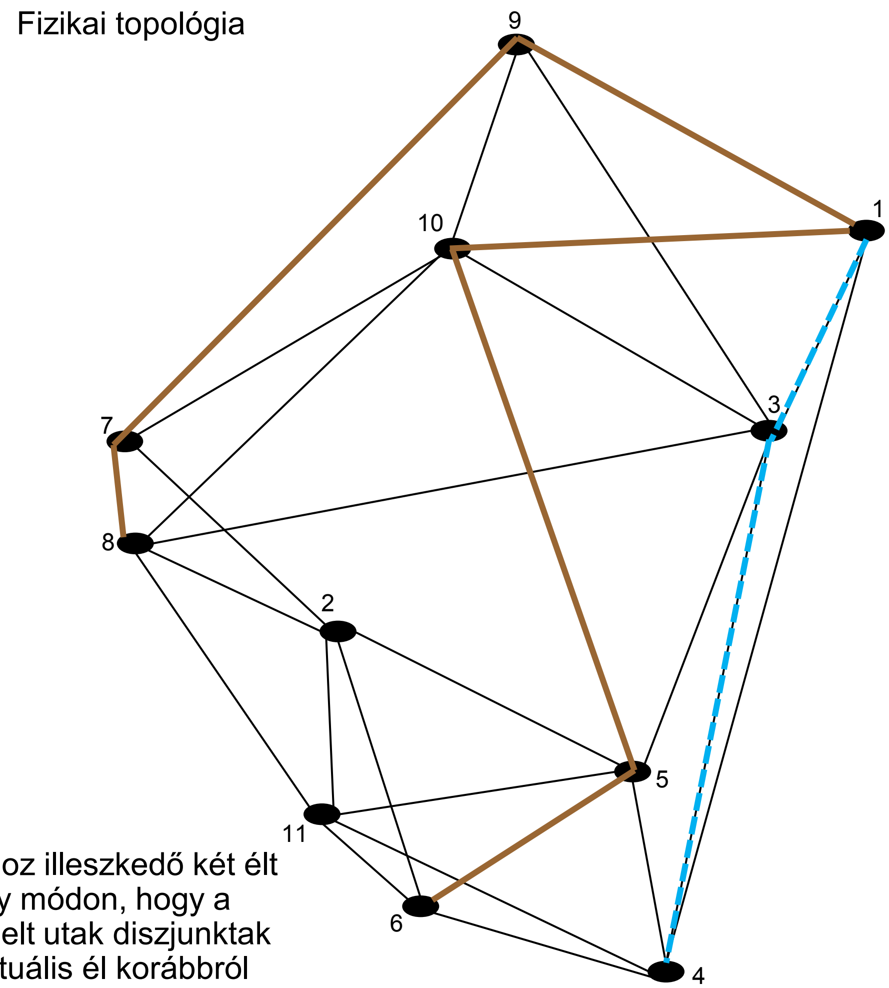


… beillesztünk egy virtuális élt.

Gráf-elméleti megközelítés

Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására



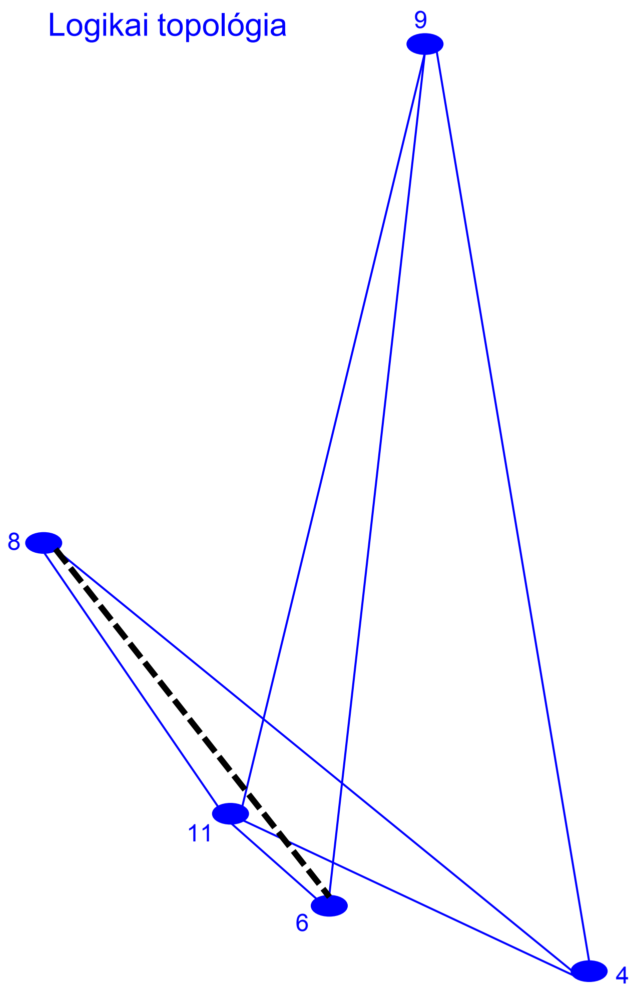


A 1-es csúcshoz illeszkedő két élt leképezzük oly módon, hogy a hozzájuk rendelt utak diszjunktak legyenek a virtuális él korábbról meghatározott útjától.

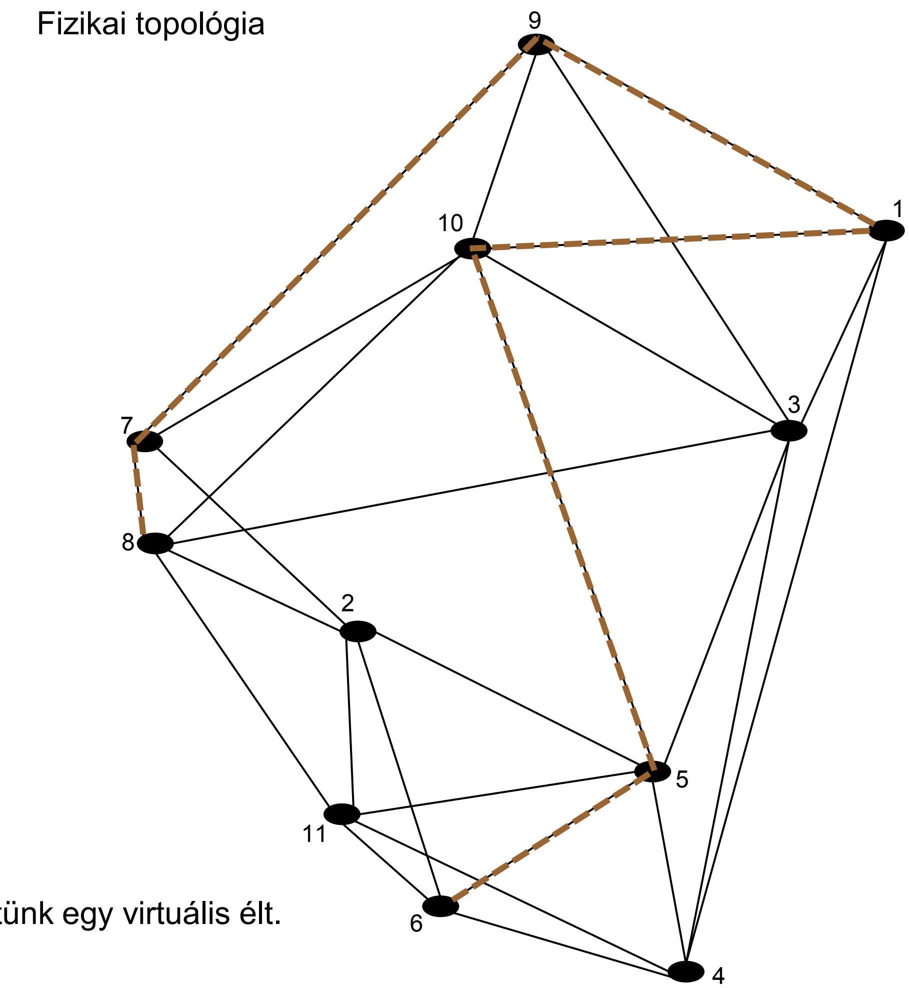
Mivel az egyszerű csúcs-elhagyás nem alkalmazható …

Gráf-elméleti megközelítés

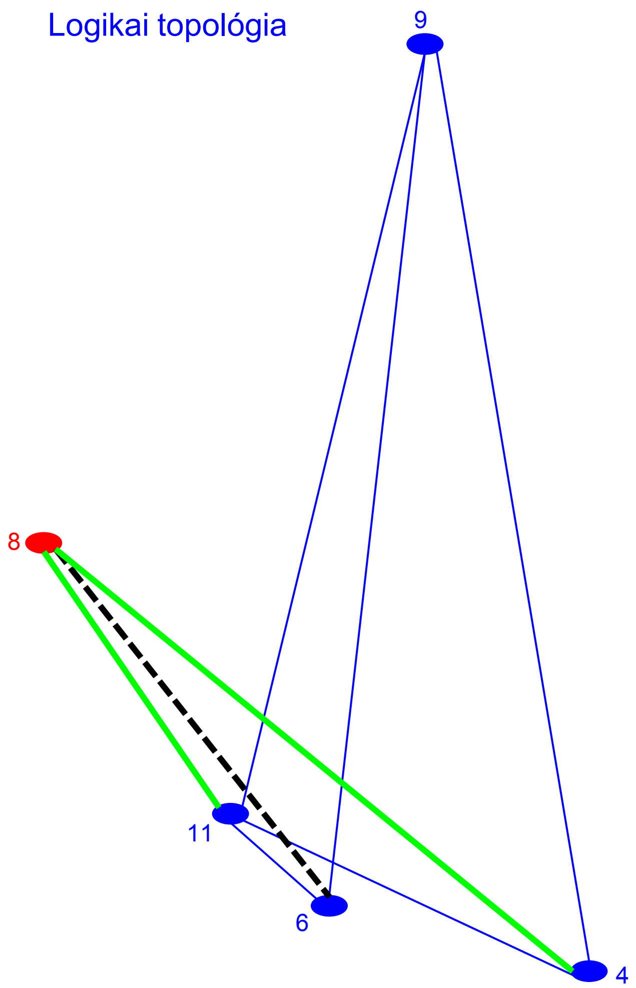
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

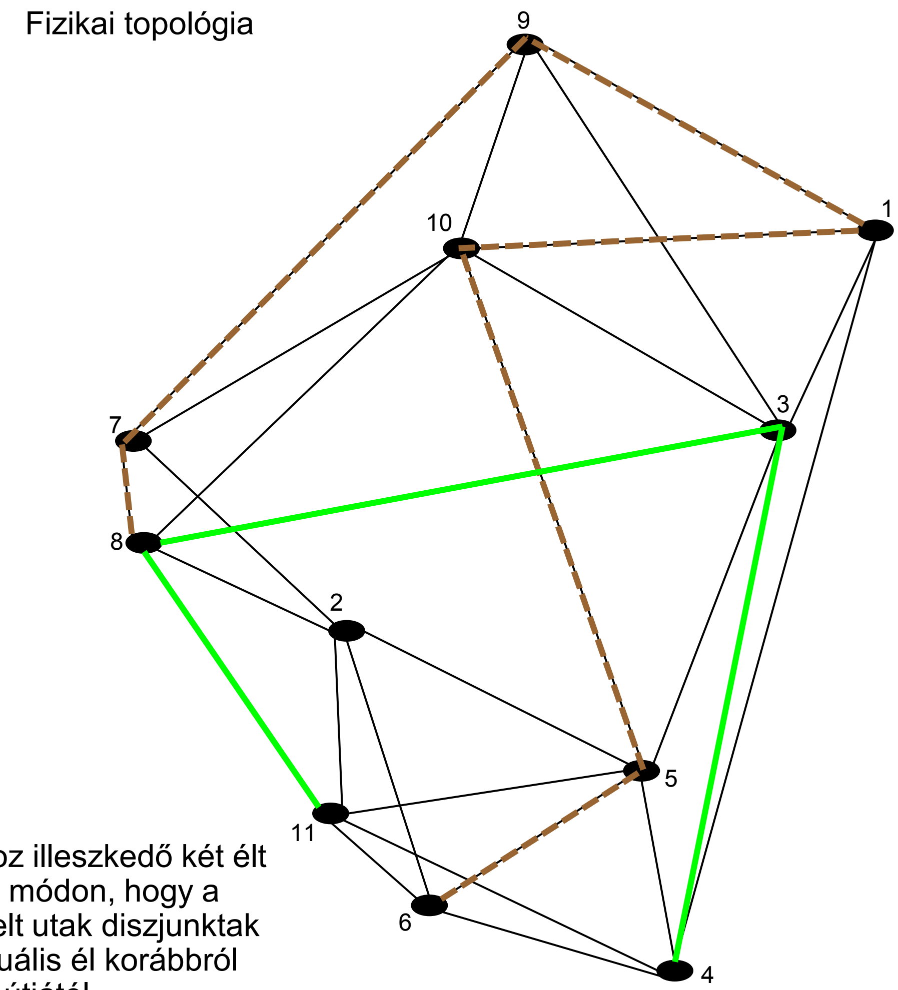


… beillesztünk egy virtuális élt.



Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására



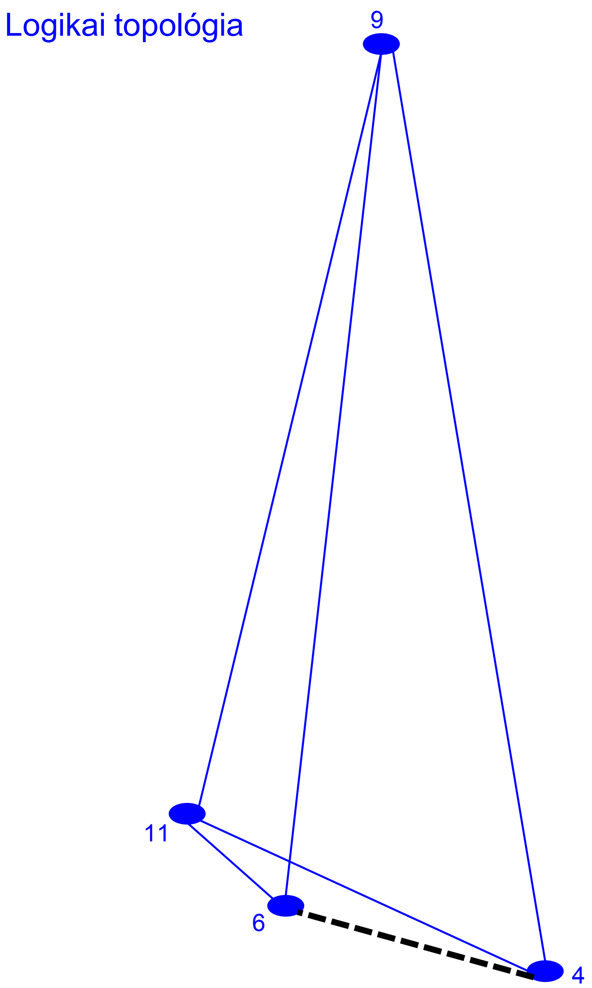


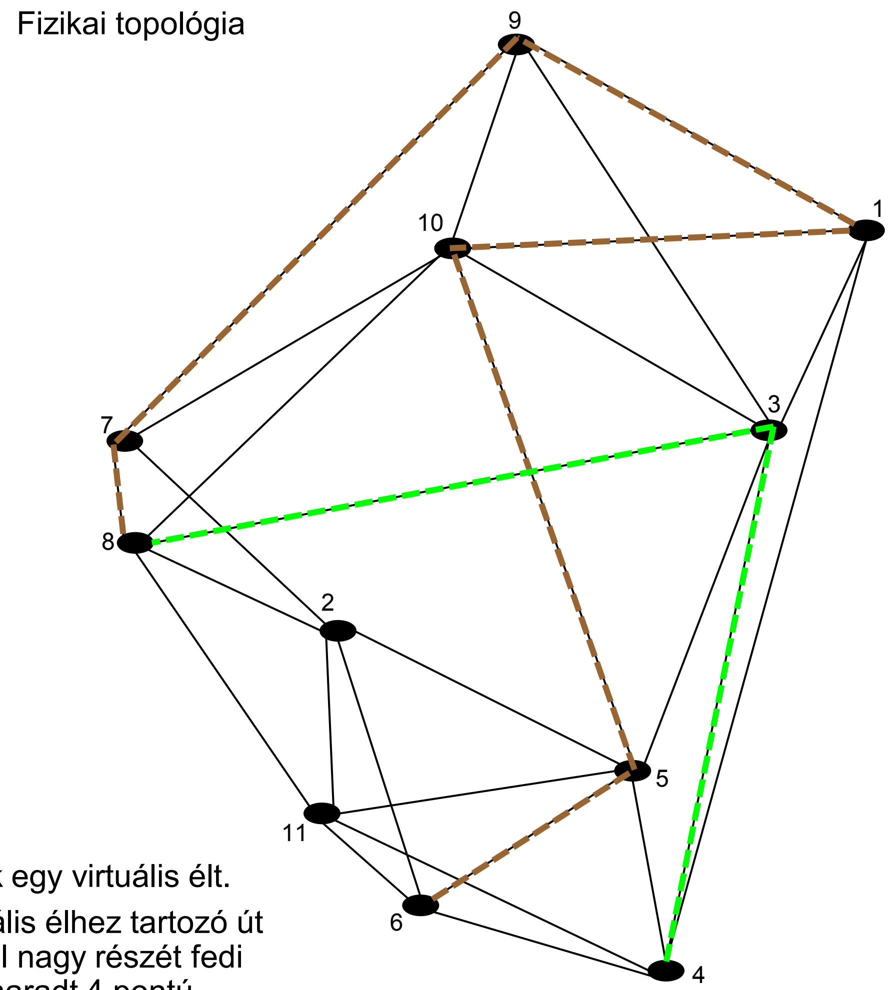
A 8-as csúcshoz illeszkedő két élt leképezzük oly módon, hogy a hozzájuk rendelt utak diszjunktak legyenek a virtuális él korábbról meghatározott útjától.

Mivel az egyszerű csúcs-elhagyás nem alkalmazható …

Gráf-elméleti megközelítés

Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

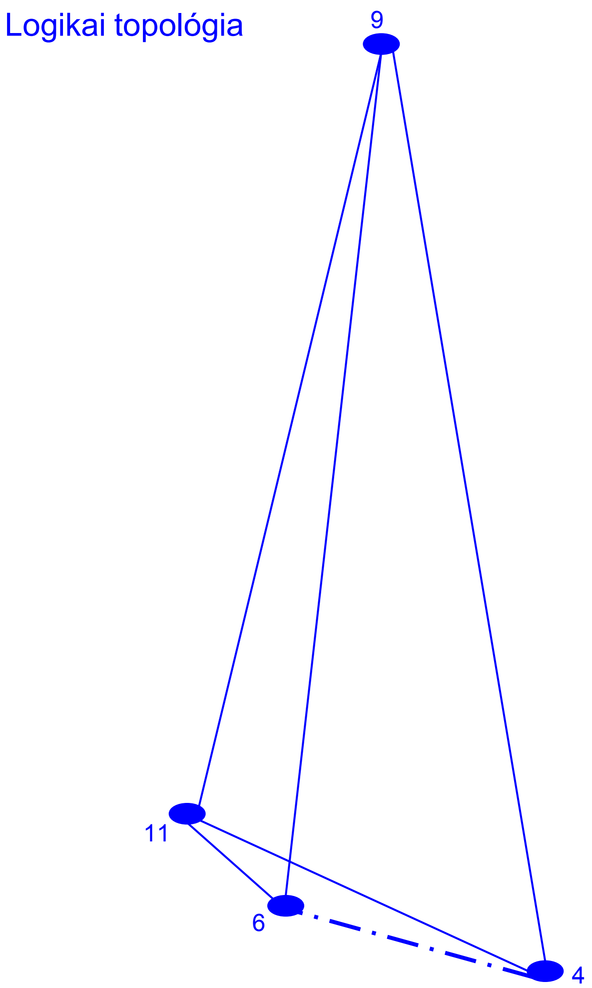


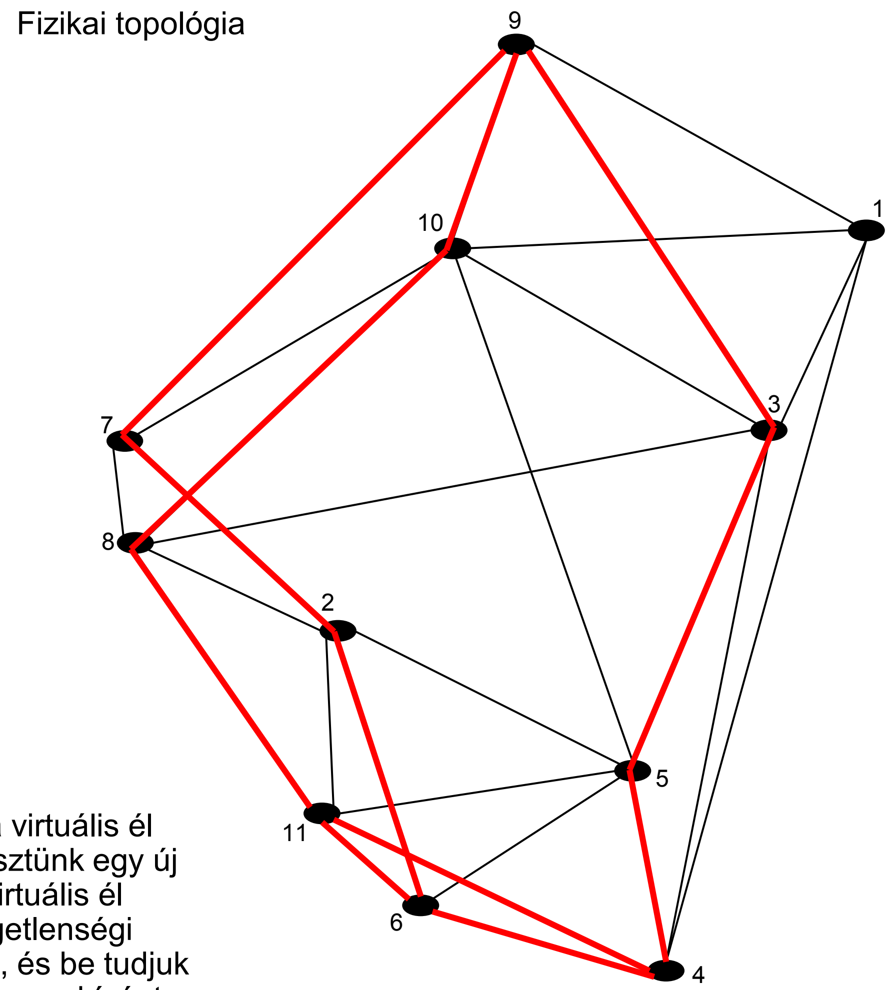


… beillesztünk egy virtuális élt.

Sajnos a virtuális élhez tartozó út a fizikai gráf túl nagy részét fedi le, így a megmaradt 4 pontú gráfot nem tudjuk a feltételeknek megfelelően leképezni.

Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

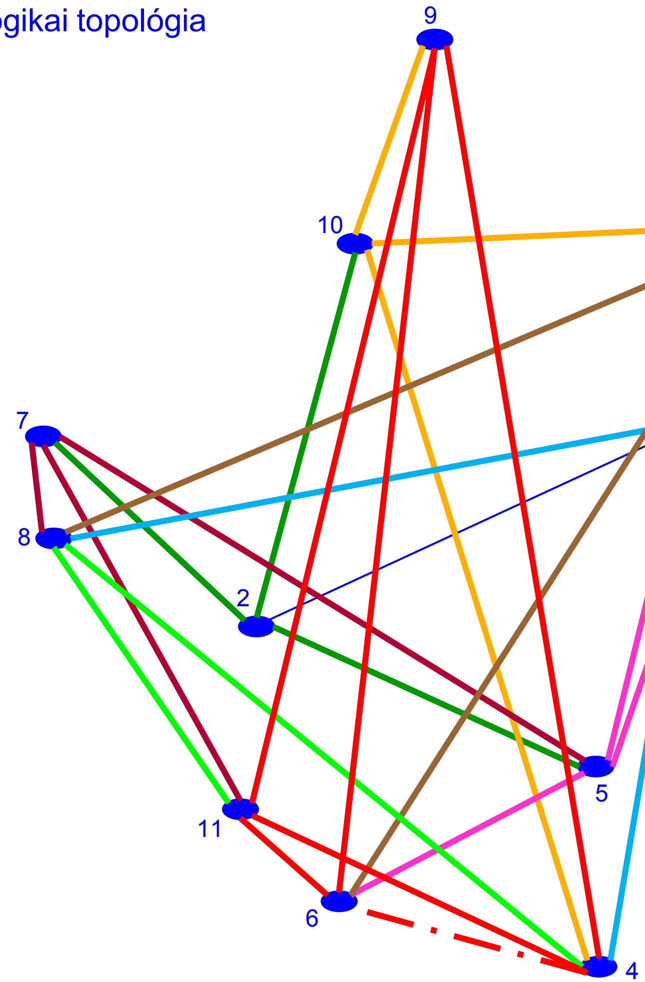




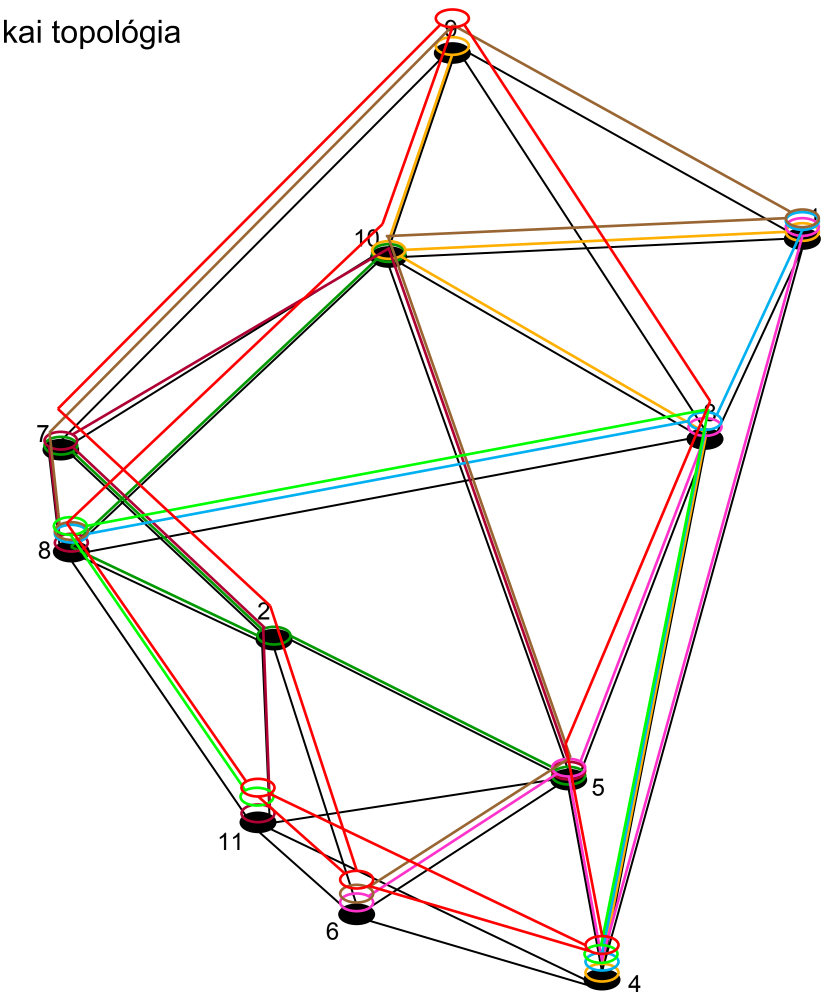
Ha azonban a virtuális él helyére beillesztünk egy új logikai élt, a virtuális él jelentette függetlenségi feltétel eltűnik, és be tudjuk fejezni az élek megkívánt leképezését.

Gráf-elméleti megközelítés
Kétszeres csúcs-hibák esete – példa az algoritmus alkalmazására

Logikai topológia



Fizikai topológia



Gráf-elméleti megközelítés
A gráfelméleti heurisztika előnyei csúcs-hibák esetén

- Ha nem is találja meg a teljes logikai topológia leképezését, a kapott parciális megoldás is használható.
- Ha sikertelen is a leképezési kísérlet, a kapott (rész-) eredmény segítséget nyújt a szükséges bővítések (logikai élek hozzáadása) helyének meghatározásához.
- ~~Többszörös csúcs-hibák esetére is általánosítható.~~

Lehetőségek az NSN-nél

- Ipari konzulens
  - Témalabor
  - TDK
  - BsC
  - MsC
  - PhD
- Projektek
  - Analitikus modellezés: Sztochasztika Tanszék, Balázs Márton
  - Szimulációs teljesítmény-analízis (C++ tudás szükséges)
  - Probléma detektálás/diagnózis rádiós hálózatokban adatbányászati eszközök felhasználásával

Kapcsolat: Radics Norbert (norbert.radics@nsn.com)

Nokia Siemens Networks