Hálózattervezés Alapjai
2007

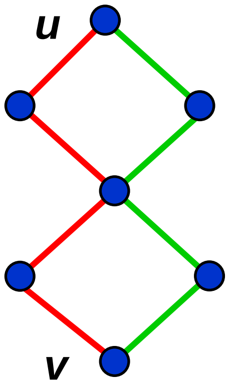
10: Hálózatok hibatoleranciája I

Hálózatok hibatoleranciája

- Milyen következményekkel jár egy csomópont (router) vagy egy él (üvegkábel) kiesése?
- Egy fontos szempont a hálózattopológia tervezésénél, hogy a hálózat ilyen esetben is működőképes maradjon: ne legynek a hálózatnak olyan részei, amelyek el vannak vágva más részektől.

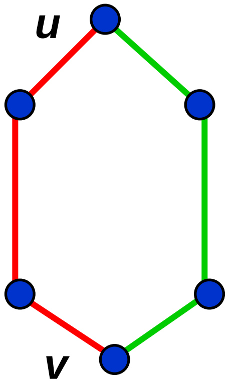
Gráfok csomópont- és élösszefüggősége

- Egy irányítatlan gráf $G=(V,E)$ *k*-szorosan élösszefüggő (*k*-edge connected), ha minden $u,v\in V$, $u\neq v$, csomópontpárra legalább *k* éldiszjunkt út van *G*-ben. A *G* gráf élösszefüggősége az a maximális *k*, amelyre *G* *k*-szorosan élösszefüggő.



2-szeresen élösszefüggő gráf

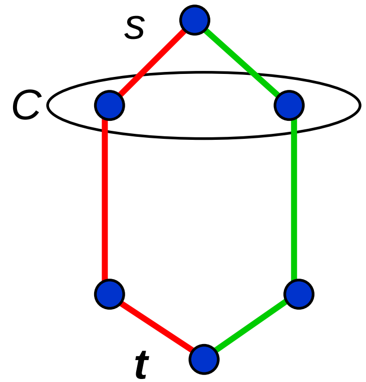
- Egy irányítatlan gráf $G=(V,E)$ *k*-szorosan csomópontösszefüggő (*k*-vertex connected), ha minden $u,v\in V$, $u\neq v$, csomópontpárra legalább *k* csomópontdiszjunkt út van *G*-ben. A *G* gráf csomópontösszefüggősége az a maximális *k*, amelyre *G* *k*-szorosan csomópontösszefüggő.



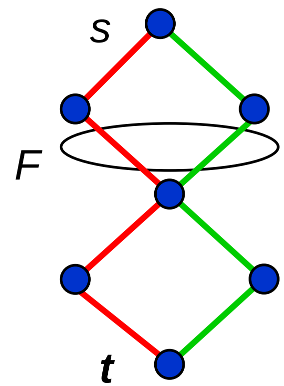
2-szeresen csomópontösszefüggő gráf

Gráfok csomópont- és élösszefüggősége

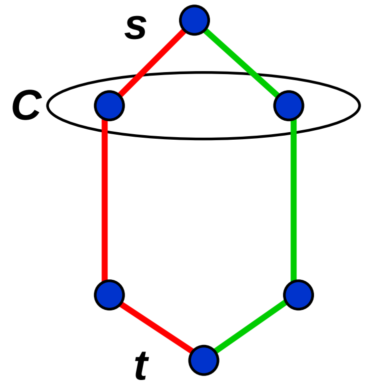
- Egy $G=(V,E)$ irányítatlan gráfban két különböző s és t csomóponthoz egy $C \subseteq V \setminus \{s,t\}$ halmazt *st*-szeparáló csomóponthalmaznak nevezünk, ha $G \setminus C$ nem tartalmaz utat s-től t-hez. Egy $C \subseteq V$ halmazt szeparáló csomóponthalmaznak (vertex separator) nevezünk, ha C valamely s és t csomópontpárra egy *st*-szeparáló csomóponthalmaz.



- Egy $G=(V,E)$ irányítatlan gráfban két különböző csomóponthoz egy $F \subseteq E$ halmazt *st*-szeparáló élhalmaznak nevezünk, ha $G \setminus F$ nem tartalmaz utat s-től t-hez. Egy *st*-szeparáló élhalmazt *st*-vágásnak (*st*-cut) is neveznek. Egy $F \subseteq V$ halmazt szeparáló élhalmaznak (edge separator) vagy vágásnak (cut) hívunk, ha F valamely s és t csomópontra egy *st*-szeparáló élhalmaz.

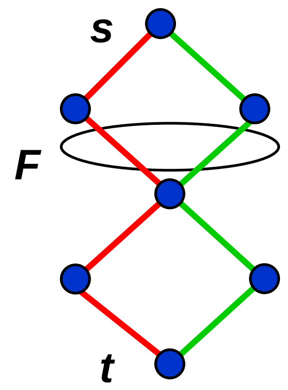


Gráfok csomópont- és élösszefüggősége



Tétel (Menger 1927)

- Egy $G=(V,E)$ irányítatlan gráfban két nem szomszédos *s* és *t* csomópontra a csomópontdiszjunkt utak maximális száma *s*-től *t*-hez egyenlő a minimális *st*-szeparáló csomóponthalmaz méretével.
- Egy $G=(V,E)$ irányítatlan gráfban két különböző *s* és *t* csomópontra az éldiszjunkt utak maximális száma *s*-től *t*-hez egyenlő a minimális *st*-szeparáló élhalmaz méretével.



Gráfok csomópont- és élösszefüggősége

Tétel (Whitney 1932)

- Egy irányítatlan gráf G legalább k+1 csomóponttal pontosan akkor k-szorosan csomópontösszefüggő, ha minden szeparáló csomóponthalmaz legalább k csomópontot tartalmaz.
- Egy irányítatlan gráf G pontosan akkor k-szorosan élösszefüggő, ha minden szeparáló élhalnaz legalább k élt tartalmaz.

Megjegyzés: Ahhoz, hogy egy kommunikációs hálózat k tetszőleges csomópont (él) kiesése után összefüggő maradjon, a hálózat-topológiának (k+1)-szeresen csomópontösszefüggőnek (élösszefüggőnek) kell lenni.

Gráfok csomópont- és élösszefüggősége

Algoritmikus, Gráfelméleti Problemák:

- Egy adott *G* gráfhoz és egy *k* értékhez döntsük el, hogy *G* *k*-szorosan csomópontösszefüggő-e (élösszefüggő-e).
- Határozzuk meg egy adott *G* gráfhoz azt a legnagyobb *k*-t, amelyre *G* *k*-szorosan csomópontösszefüggő (élösszefüggő).
  Határozzunk meg egy minimális szeparáló csomóponthalmazt (élhalmazt).
- Connectivity Augmentation: Határozzunk meg egy adott *k* értékhez és egy *G* gráfhoz (amely nem *k*-szorosan csomópont- élösszefüggő) egy minimális *F* élhalmazt, úgy hogy *G* U *F* *k*-szorosan csomópontösszefüggő (élösszefüggő).
  - Élösszefüggőség esetén a probléma polinomiális időben megoldható minden *k*-ra.
  - Csomópont-összefüggőségnél csak $k \leq 4$ esetben ismert polinomiális idejű algoritmus
  - A súlyozott változat (ahol minden lehetséges élnek van egy súlya) csomópont- és élösszefüggőségre is *NP*-nehéz már $k = 2$ esetén is.

Minimális vágás (Min-Cut)

- Kiszámítunk egy minimális vágást gráfokban, ahol az élek súlyozattak.
- A minimális vágás (Min-Cut) $G=(V,E)$-ben $c : E \to \mathrm{R}^+$ élsúlyokkal egy olyan F élhalmaz, hogy $G \backslash F$ nem összefüggő és $c(F)=\Sigma_{e \in F} c(e)$ minimális.
- Az élösszefüggőség kiszámításához minden él súlyát 1-re állítjuk, ekkor egy vágás súlya egyenlő a vágás éleinek számával

Probléma MinCut

- Adott: irányítatlan $G=(V,E)$ gráf $c : E \to \mathrm{R}^+$ élsúlyokkal.
- Megoldás: Élhalmaz F, úgy hogy $G \backslash F$ nem összefüggő.
- Cél: minimalizáljuk $c(F)=\Sigma_{e \in F} c(e)$.

Minimális vágás **[Nagamochi, Ibaraki 92] [Stoer and Wagner 97]**

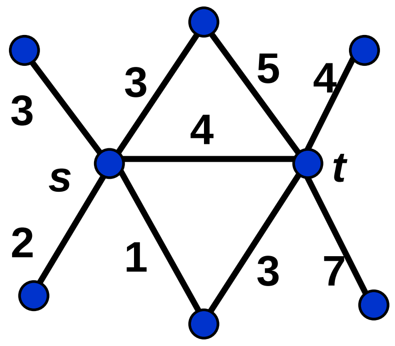
Alapötlet:

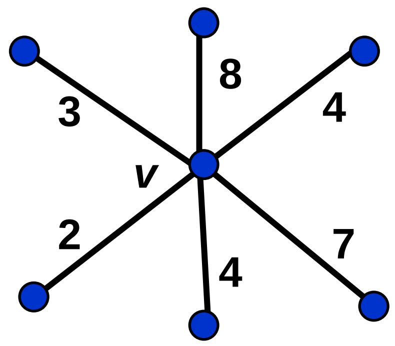
- Először kiszámítunk egy minimális *st*-vágást C_1-t két tetszőleges *s* és *t* csomóponthoz.
- Ezután összeolvasztjuk *s*-t és *t*-t és az így keletkezett gráfban rekurzívan kiszámítunk egy C_2 minimális vágást.
- A végén a két vágás C_1 és C_2 közül azzal a vágással tér vissza az algoritmus, amelyiknek a súlya alacsonyabb.

Minimális vágás

Az *s* és *t* csomópontok összeolvasztása egy új *v* csomóponttá:

- Ha *s* és *t* egy *e* élel egymassal adjacens volt, akkor *e*-t töröljük.
- Ha *s* egy *e* éllel és *t* egy *e*´ éllel egy $w \neq s,t$ csomóponttal adjacens volt, akkor a *v* és *w* közötti él súlya $c(e)+c(e´)$ lesz.
- Ha *s* egy e éllel egy $w \neq s,t$ csomóponthoz adjacens volt, de *t* és *w* nem voltak adjacensek, akkor a *v* és *w* közötti él súlya $c(e)$ lesz.
- Ha *t* egy e éllel egy $w \neq s,t$ csomóponthoz adjacens volt, de *s* és *w* nem voltak adjacensek, akkor a *v* és *w* közötti él súlya $c(e)$ lesz.

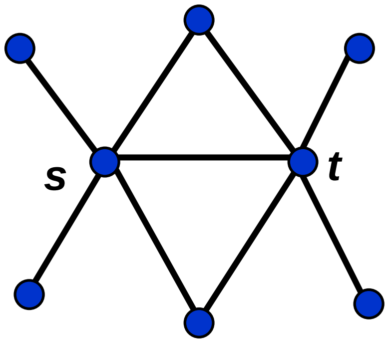




Minimális vágás

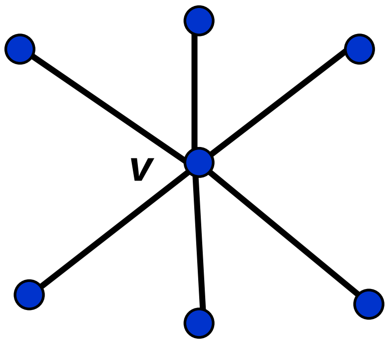
Legyen *H* egy gráf, amely *G*-ből csomópont-összeolvasztások után állt elő.
Azt mondjuk, hogy egy *H* beli *w* csomópont tartalmaz egy *G* beli *v* csomópontot, ha

- *w*=*v*, vagy
- *w* a *w*´ és *w*´´ csomópontok összeolvasztásával állt elő, és *w*´ vagy *w*´´ tartalmazta *v*-t.



Azt mondjuk, hogy egy *H* beli (*w*,*z*) él tartalmaz egy *G* beli (*u*,*v*) élt, ha *w* tartalmazza *u*-t és *z* tartalmazza *v*-t, vagy *w* tartalmazza *v*-t és *z* tartalmazza *u*-t.

Minden *F* vágás *H*-ban megfelel egy vágásnak *G*-ben, úgy hogy minden $e \in F$ élt helyettesítünk azon *G* beli élekkel, amelyeket *e* tartalmaz. A *H* beli vágást így felfoghatjuk úgy mint egy vágás *G*-ben.



v tartalmazza *s*-t és *t*-t

Minimális vágás

Lemma 1: Legyen s és t két csomópont G-ben, $s \neq t$, és legyen H az a gráf, amely G-ből s és t összeolvasztása után előáll.
Legyen C_1 egy minimalis st-vágás G-ben és C_2 egy minimalis vágás H-ban.
Ekkor az a vágás, amely a kettő közül kisebb súlyú, minimális vágás G-ben.

Biz.: A vágások H-ban pontosan azoknak a G beli vágásoknak felelnek meg, melyek s-t és t-t nem szeparálják el egymástól.
Ha G-ben létezik egy minimális vágás, amely egy st-vágás, akkor C_1 egy minimális vágás G-ben.
Ha nincs olyan minimális vágás, akkor a minimális vágás nem szeparálja el s-t és t-t, ezét C_2 egy minimális vágást ad G-ben. □

Minimális vágás – Az algoritmus

A Min-Cut algoritmus:

- Az algoritmus $|V|$-1 fázisban dolgozik.
- Az i-edik fázisban kiszámít egy C_i vágást, amely két bizonyos s és t csomóponthoz az aktuális gráfban egy st-vágás.
  Ezután összeolvasztja s-t és t-t.
- Az algoritmus azt a C_j vágást adja outputként, amely a $|V|$-1 kiszámított vágás közül minimális súlyú, azaz, amelyre

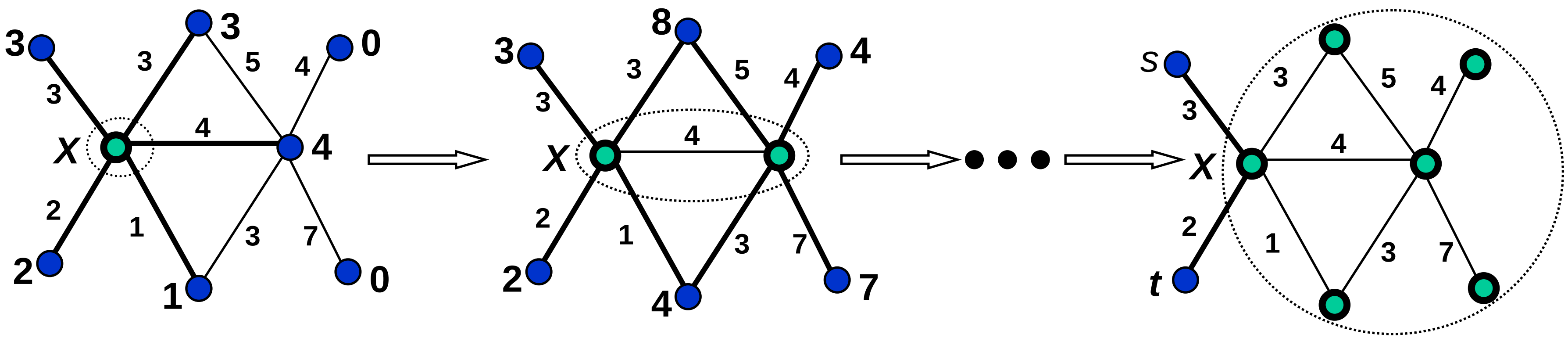
$$c(C_j) = \min_{1 \leq i < |V|} \{c(C_i)\}.$$

Hogyan lehet egy fázisban az s és t csomópontot ügyesen választani?

Minimális vágás – Az algoritmus egy fázisa

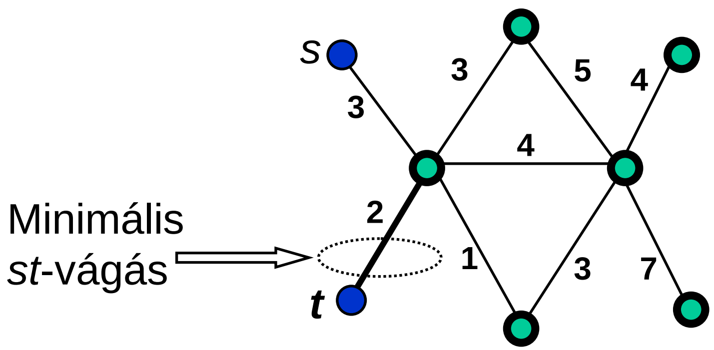
Tekintsük az *i*-edik fázist:

- Legyen $H=(V',E')$ az aktuális gráf a fázis elején.
- Csomópontok egy $X \subset V'$ részhalmazára és egy $y \in V' \setminus X$ csomópontra jelölje $c(X,y) = \Sigma_{x \in X, \{x,y\} \in E'} c(\{x,y\})$ mindazon élek súlyának összegét, amelyek *y*-t egy $x \in X$ csomópontottal kötnek össze.
- Kezdetben válasszunk egy tetszőleges $x \in V'$ csomópontot és legyen $X=\{x\}$. Amíg $X \neq V'$, az algoritmus azt a $z \in V' \setminus X$ csomópontot adja hozzá *X*-hez, amelyre $c(X,z)$ maximális.
- Legyen *s* az utolsóelőtti és *t* az utolsó csomópont, amelyet az *i*-edik fázisban *X*-hez hozzáadunk.



Minimális vágás – Az algoritmus egy fázisa

- Legyen *s* az utolsóelőtti és *t* az utolsó csomópont, amelyet az *i*-edik fázisban *X*-hez hozzáadunk.
- Ekkor a *t*-hez *H*-ban incidens élek halmaza adja meg a C_i vágást (helyesség bizonyítása később következik).
  C_i-t *G* beli vágásként interpretálva, C_i azon *G* beli élekből áll, amelyek egy *t*-ben tartalmazott csomópontot egy *t*-ben nem tartalmazott csomóponttal kötnek össze.

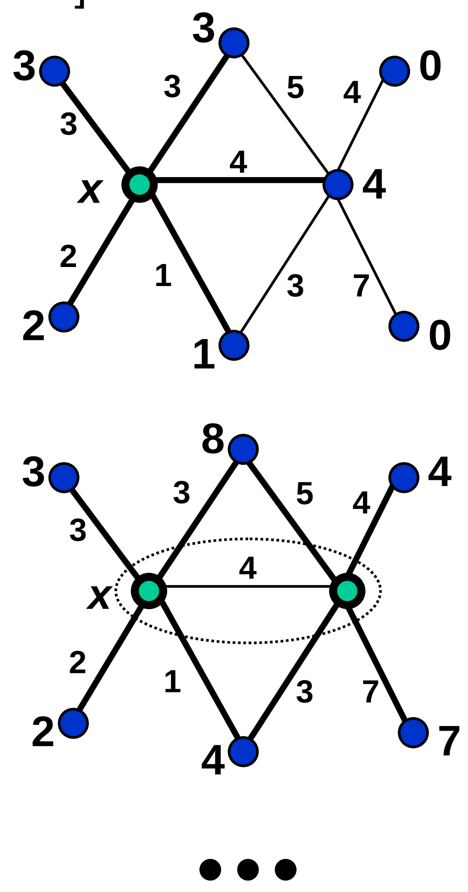


Minimális vágás – Az algoritmus egy fázisa

Algoritmus One_Phase_of_the_Min_Cut_Algorithm [Stoer, Wagner 97]

- Input: irányítatlan $H=(V', E')$ gráf $c: E \to \mathbb{R}^+$ élsúlyokkal.
- Output: A t csomópont, amelyhez incidens élek egy minimalis st-vágást adnak meg H-ban.

```
1.   Legyen x ∈V´ egy tetszőleges csomópont; legyen X:={x};
2.   Legyen Q egy Priority-Queue a csomópontokhoz;
     // Invariáns: Q pontosan V´\X csomópontjait tartalmazza és
     //   v∈Q ⇒ Q.Priority(v) = c(X,v)
3.   for all  e={x,v}  do  Q.Insert(v, c(e));  od;
4.   for all  v ∈ V´\{x} melyre v ∉ Q  do  Q.Insert(v, 0);  od;
5.   t := x;
6.   while Q ≠ Ø do
7.       t := Q.DeleteMax( );
8.       X := X U {t};
9.       for all e={t,w} do
10.          if w∈Q then  Q.IncreasePriority(w, Q.Priority(w)+c(e)); fi;
11.      od;
12.  od;
13.  return t;
```



Minimális vágás – Futási idő

A Min-Cut algoritmus futási ideje:

- Egy fázis: $O(|V| \log |V| + |E|)$ idő (Fibonacci-Heap-pel).
  - $|E'|$ *Insert*: amortizálva $O(1)$ idő operációnként
  - $|V'|-1$ *DeleteMax*: amortizálva $O(\log|V'|)$ idő operációnként
  - $\leq|E'|$ *IncreasePriority*: amortizálva $O(1)$ idő operációnként

- Összesen $|V|-1$ fázis: $O(|V|^2 \log|V| + |V||E|)$ idő.

Minimális vágás - Helyesség

Meg kell mutatni, hogy az *i*-edik fázisban kiszámított C_i vágás az ebben a fázisban *X*-hez két utolsónak hozzáadott csomóponthoz *s*-hez és *t*-hez egy minimális *st*-vágás.

Lemma 2: Legyen $H=(V',E')$ a gráf egy fázis elején. Legyen *s* és *t* a két csomópont, amelyet ebben a fázisban utolsónak adtunk hozzá *X*-hez. Ekkor a *t*-hez incidens élek egy minimális *st*-vágást képeznek *H*-ban.

Biz.: Legyen $F = \{ \{t,v\} \in E' \}$ a *t*-hez incidens élek halmaza és C egy tetszőleges *st*-vágás *H*-ban. Megmutatjuk, hogy

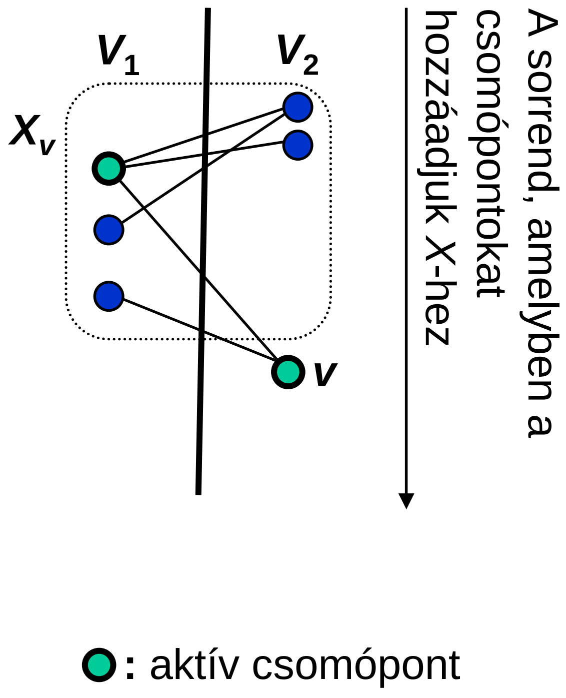
$$c(C) \geq c(F).$$

Ebből következik, hogy *F* egy minimális *st*-vágás.

Minimális vágás - Helyesség

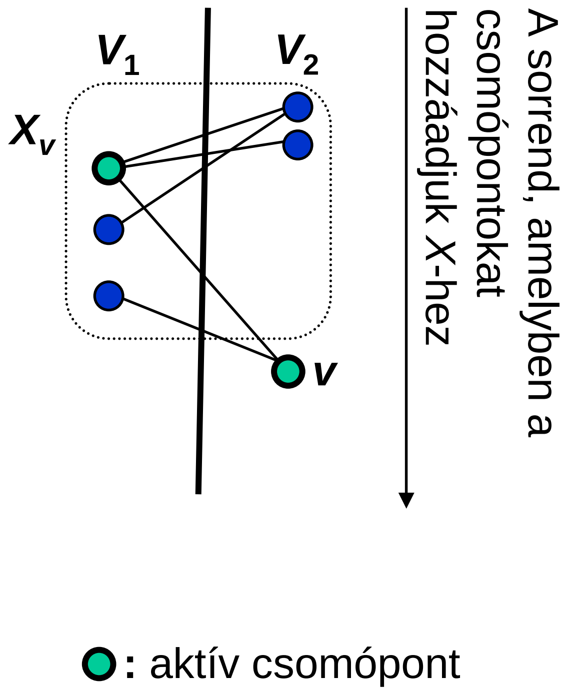
- Legyen V_1 és V_2 a két csomóponthalmaz, úgy hogy C pontosan azon élek halmaza, melyek V_1 és V_2 között vannak.
- Egy $v \neq x$ csomópontot aktivnak hívunk, ha v és a közvetlenül v előtt X-hez hozzáadott csomópont különböző V_i-ben (i=1,2) van.
- Jelölje X_v azon csomópontok halmazát, amelyeket v előtt adtuk hozzá X-hez. C_v azt a vágást, amelyet C-ben az X_vU{v} csomóponthalmaz indukál, azaz

$$C_v = C \cap \{ \{a,b\} : a,b \in X_v\text{U}\{v\} \}.$$



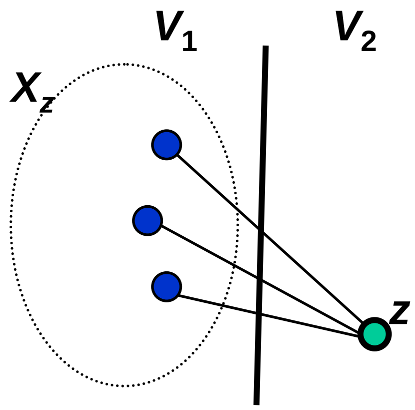
Minimális vágás - Helyesség

- Legyen C_v az a vágás, amelyet C-ben az $X_v \cup \{v\}$ csomóponthalmaz indukál, azaz
  $$C_v = C \cap \{ \{a,b\} : a,b \in X_v \cup \{v\} \}.$$
- Megmutatjuk indukcióval az aktív csomópontok számán, hogy minden v aktív csomópontra teljesül, hogy
  $$c(X_v,v) \leq c(C_v). \qquad (1)$$
- A t csomópont nyilvánvalóan aktív csomópont:
  s és t különböző V_i-ben (i=1,2) van és s a közvetlenül t előtt X-hez hozzáadott csomópont.
  Ekkor a $c(X_t,t) \leq c(C_t)$ egyenlőtlenség pontosan a $c(F) \leq c(C)$ állítást adja.

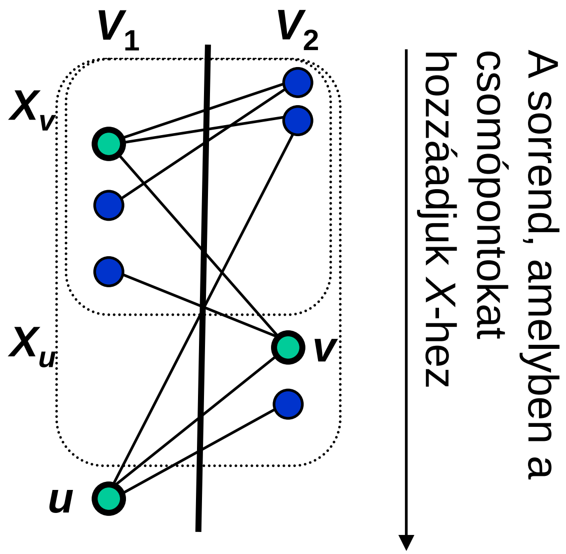


Minimális vágás - Helyesség

- Az első aktív z csomópontra
  $c(X_z,z) = c(C_z)$, tehát (1) teljesül:
  z definíció szerint az első csomópont, amelyre
  $X_z \subseteq V_i$, mondjuk $X_z \subseteq V_1$, és $v \in V_2$. Ekkor
  $$c(X_z,z) = \Sigma_{w \in X_z} c(z,w) = c(C_z),$$



- Legyen v egy aktív csomópont. Tegyük fel, hogy minden aktív csomópontra v-ig (inkluzív v):
  $$c(X_v,v) \leq c(C_v).$$
  Legyen u a következő aktív csomópont. Ekkor:
  $$c(X_u,u) = c(X_v,u) + c(X_u \backslash X_v,u).$$

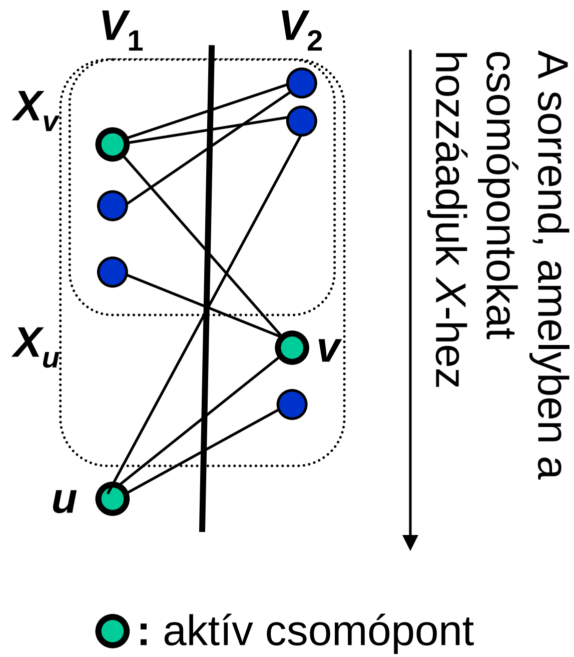


Minimális vágás - Helyesség

- Mivel az algoritmus *v*-t *u* előtt vette ki a Priority-Queue-ból, teljesül hogy
  $$c(X_v,u) \le c(X_v,v).$$
  Az indukciós feltételből következik, hogy
  $$c(X_v,v) \le c(C_v).$$
  Ezért
  $$c(X_u,u) \le c(C_v) + c(X_u \backslash X_v,u). \qquad (2)$$

- Minden él $X_u \backslash X_v$ és *u* között benne van a C_u vágásban, de nincs benne C_v-ben. Ezért
  $$c(C_v) + c(X_u \backslash X_v,u) \le c(C_u). \qquad (3)$$

- (2)-ből és (3)-ból következik az indukciós állítás:
  $$c(X_u,u) \le c(C_u). \qquad \square$$



Minimális vágás

Tétel 1: A bemutatott Min-Cut algoritmus $O(|V|^2 \log|V| + |V||E|)$ időben kiszámít egy minimális vágást egy inrányítatlan gráfban pozitív élsúlyokkal.

Biz.: A helyességet indukcióval bizonyítjuk a gráf csomópontjainak számán.
$|V|=2$ esetén az algoritmus helyesen számítja ki a minimális vágást.
Tegyük fel, hogy az algoritmus minden legfeljebb (n-1) csomópontú gráfra helyesen dolgozik. Legyen G egy n csomópontú gráf.
Lemma 2 szerint az algoritmus az első fázisban helyesen kiszámít egy minimális *st*-vágást.
Legyen H az a gráf, amely G-ből s és t összeolvasztásával áll elő.
Az indukciós feltétel szerint a további fázisokban kiszámított vágások legkisebbike egy minimális vágás H-ban.
Ekkor Lemma 1 szerint az első és a további fázisokban kiszámított vágások legkisebbike egy minimális vágás G-ben. □

Irodalom

[Nagamochi, Ibaraki 92]: H. Nagamochi and T. Ibaraki. **Computing Edge-Connectivity in Multigraphs and Capacited Graphs**. *SIAM Journal of Discrete Mathematics*, Vol 5(1), 54-66, 1992.

[Stoer, Wagner 97]: M. Stoer and F. Wagner. **A simple Min-Cut Algorithm**. *Journal of the ACM*, Vol. 44(4), 585-591, 1997.