

# Javítgatásos algoritmusok, Ford—Fulkerson-algoritmus

Hajnal Péter

Bolyai Intézet, TTIK, SZTE, Szeged

2020. ősz

# Alapfogalmak

## Definíció: Hálózat

Legyen  $\vec{G}$  irányított gráf,  $s, t \in V(G)$  két KÜLÖNBÖZŐ kijelölt csúcs és  $c : E(G) \rightarrow \mathbb{R}_{++}$  függvény. Ekkor  $(\vec{G}, s, t, c)$  négyesét *hálózatnak* nevezzük, ahol  $s$  pontot *forrásnak*,  $t$  pontot *nyelőnek*,  $c$ -t pedig *kapacitásfüggvénynek* nevezzük.

Hálózatok sok gyakorlati probléma absztrakciójához hasznosak. Például egy város vízvezetékhalózata írható így le, ahol a kapacitásfüggvény a csövek terhelhetőségét (például átmérő) adja meg. Egy úthálózat is modellezhető így. Egy él kapacitása az áteresztő képessége, a megfelelő útszakasz szélességével, sávjainak számával arányos.

# Folyamok definíciója

A fenti fogalom egy statikus fogalom. A dinamika leírásához új fogalom kell.

## Definíció: Folyam (hálózatban)

Az  $f : E(G) \rightarrow \mathbb{R}$  függvényt *folyamnak* nevezzük a  $H$  hálózatban, ha

(F1) minden  $e$  él esetén  $0 \leq f(e) \leq c(e)$

(F2) minden  $v \in V \setminus \{s, t\}$  esetén

$\sum_{e:e \in E_{be}(v)} f(e) = \sum_{e:e \in E_{ki}(v)} f(e)$ , ahol  $E_{be}(x)$  az  $x$ -be befutó élek halmaza,  $E_{ki}(x)$  az  $x$  pontból kifutó élek halmaza.

Az első feltételt megengedettségnek nevezzük.  $f$  tehát megengedett, ha a csöveken nem folyik át a kapacitásnál nagyobb, illetve negatív mennyiség. A második feltételben szereplő egyenlőségek vezérlő elvét *megmaradási törvénynek* nevezzük.

# Folyamok összehasonlítása

## Példa

Az  $f \equiv 0$  folyam egy tetszőleges hálózatban. Ekkor minden élen 0 anyagmennyiség fut.

Speciálisan minden hálózatban megadható folyam. Hogy ezek a folyamok összevethetők legyenek szükségünk van egy újabb fogalomra.

## Definíció: Folyam értéke

Egy folyam értéke  $\epsilon(f) = \sum_{e \in E_{be}(t)} f(e) - \sum_{e \in E_{ki}(t)} f(e)$ .

$t$  a nyelő.

Az  $f$  folyam értéke negatív is lehet, ilyenkor visszafele folyik a víz a csőhálózatban. Az üres folyam értéke 0.

# A folyam probléma

## Definíció: Folyam probléma

Adott egy hálózat. Keressünk hozzá egy maximális értékű folyamot.

A maximális jelző első olvasatban problémás. Egy folytonos problémával állunk szemben, amikor nem szükséges a legnagyobb érték fevétele.

Egy folyam egy  $m$  élű hálózatban  $m$  valós szám leírásával adható meg, azaz azonosítható  $\mathbb{R}^E \cong \mathbb{R}^m$  egy pontjával. A folyamatoknak megfelelő pontok  $\mathbb{R}^m$  egy kompakt halmaza, amelyen az érték egy folytonos függvény. Ez a folytonos függvény felveszi maximumát  $\mathbb{R}^{|E(G)|}$  egy kompakt részhalmazán.

Tulajdonképpen a folyam probléma a lineáris programozás feladat egy speciális esete.

# Előzetes megjegyzések

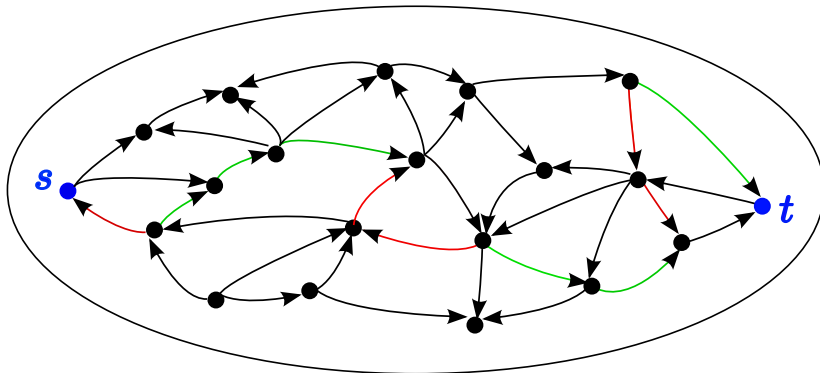
Legyen  $P$  egy irányítatlan értelemben vett út  $\vec{G}$ -ben.

Azaz hagyjuk el a  $\vec{G}$  gráf irányítását (így kapjuk a  $G$  irányítatlan gráfot).  $P$  egy út ebben.

$P$  éleit két kategóriába sorolhatjuk ( $\vec{G}$ -beli irányításának megfelelően: vagy előrehaladó él (azaz a  $P$ -t leíró pont-él-pont-él... sorozatban kiinduló végpontja előbb van) vagy hátramutató él.

Ezen élek halmazai  $E_{\text{előre}}(P)$  és  $E_{\text{hátra}}(P)$ . Így  
 $E(P) = E_{\text{előre}}(P) \cup E_{\text{hátra}}(P)$ .

## Példa



**Figure:** Egy irányítatlan  $st$  út egy hálózatban. A két kék csúcs a forrás és nyelő. Az út élei a piros és zöld élek. A zöld élek előre mutatóak a forrás  $\rightarrow$  nyelő viszonylatban, míg a piros élek hátra mutatóak.

# Javító út fogalma

## Definíció

Legyen  $H(\vec{G}, s, t, c)$  hálózat, ebben pedig egy  $f$  folyam. Ekkor **P javítóút** ( $f$  folyamra,  $H(\vec{G}, s, t, c)$  hálózatban), ha

(J1)  $P$  egy  $s$ -ből induló út  $G$ -ben.

(J2)  $P$  egy  $t$ -be vezető út  $G$ -ben.

(J3)  $e \in E_{\text{előre}}(P)$  esetén  $f(e) < c(e)$ , míg  $e \in E_{\text{hátra}}(P)$  esetén  $f(e) > 0$ , azaz ha az előrehaladó éleken nem a folyam nem használja ki a csőszakasz által engedett maximumot (az él kapacitását) valamint visszafele (a hátramutatató éleken) is történik bizonyos „visszafolyás”.

Megjegyezzük, hogy (J1) és (J2) együtt azt mondja, hogy  $P$  egy  $st$ -út  $G$ -ben.



# Miért javító?

## Lemma

Legyen  $f$  egy folyam a  $H(\vec{G}, s, t, c)$  hálózatban. Ekkor ha találunk egy  $P$  javító utat, akkor  $f$  folyam javítható, azaz nem maximális értékű.

# Bizonyítás

- Legyen  $P$  egy javító út  $f$ -re.
- Legyen  $\min_{e \in E_{\text{előre}}(P)} (c(e) - f(e)) = \delta_{\text{előre}}$  (azaz, egy élhez tartozó szám azt mutatja meg, hogy legfeljebb mennyivel növelhető az anyagmennyiség ezen élen a kapacitás korlát megsértése nélkül),
- hasonlóan  $\delta_{\text{hátra}} = \min_{e \in E_{\text{hátra}}} f(e)$ ,
- valamint a minimumuk:  $\delta = \min\{\delta_{\text{hátra}}, \delta_{\text{előre}}\}$ .
- Javító út esetén  $\delta > 0$ , azaz lehet még növelni az anyagáramlást.
- Legyen a **módosított folyam**

$$f^+(e) = \begin{cases} f(e), & e \notin E(P), \\ f(e) + \delta, & e \in E_{\text{előre}}(P), \\ f(e) - \delta, & e \in E_{\text{hátra}}(P). \end{cases}$$

# Bizonyítás (folytatás)

A következő észrevételek szolgálják a bizonyítás alapját:

- (1)  $\delta > 0$ .
  - (2)  $f^+$  megengedett,
  - (3)  $f^+$  teljesíti a megmaradási törvényt minden nem-nyelő,  
nem-forrás csúcsban,
  - (4)  $\epsilon(f^+) = \epsilon(f) + \delta$ .
- (2) és (3) együtt igazolja, hogy  $f^+$  egy folyam. (1) és (4) együtt igazolja, hogy  $f^+$  értéke nagyobb mint  $f$ -é.

# Szünet



# A séma

## Ford—Fulkerson-algoritmus

- (I) **Inicializálás:** Vegyünk egy kiinduló  $f$  folyamatot.  
// Például a  $f = 0$  folyamat (minden élen 0 anyagmennyiség folyik).
- (K) **Keresés:** Keressünk javító utat  $f$ -re. Ha találunk, akkor (J); ha nincs, akkor (S).
- (J) **Javítás:** A korábbi lemma alapján „javítsuk”  $f$ -et:  $f \leftarrow f^+$ .  
Vissza a (K) lépéshez.
- (S) **Stop:** Álljunk le, az aktuális folyam nem javítható Ford—Fulkerson-módon. Valamilyen értelemben „nagy értékű”.

# Kérdések

Több kérdés felmerül a fenti algoritmus leírásnál:

- (1) Hogyan keressünk javító utat adott  $f$ -re. Másképpen az eredeti feladatot (optimális folyam keresését) lecseréltük egy új problémára (javító út keresésére). Jó „üzlet” volt?
- (2) Az algoritmus struktúrája megengedi a végtelen javítássorozatot. Lehet ciklizálás? Ha van ciklizálás akkor a kiszámolt folyamsorozat értékei egy monoton növvő korlátos sorozatot adnak, azaz konvergál az érték. A limesz megegyezik a legnagyobb folyamértékkel?
- (3) Mi a viszony a nem javítható folyamatok és a maximális folyamatok között?

(1)

A Keresés lépés az egyetlen algoritmikusan problémás lépés.

Megoldása viszonylag egyszerű. Maga ez a része az algoritmusnak is javítgatásos.

### Javító út kezdemény

Egy  $\vec{G}$ -ben egy  $P$  irányítatlan út *javító út kezdemény*, ha teljesíti a (J1) és (J3) feltételt. Azaz csak az a feltétel hiányzik a javító útságból, hogy  $t$ -be/a nyelőbe vezessen.

A  $P_0 : s$  0 hosszú út egy javító út kezdemény.

Ilyen javító út kezdeményeket keresünk és próbáljuk hosszabbítani/javítani őket, remélve, hogy az egyik próbálkozás  $t$ -be vezet (és így javító utat találtunk).

# A Ford-Fulkerson-algoritmus javító út keresése

## Ford-Fulkerson javító út keresése

Keresés inicializálása: Legyen  $S := \{s\}$ .

//  $S$  azon csúcsok halmaza, ahová javítóút-kezdeményeket találtunk.

Javítóút-kezdemények növelése:

//  $S$  növelése.

Legyen

$$B_{\text{előre}} = \{x \in V \setminus S : \text{van olyan } y \in S, \text{ hogy } \vec{yx} \in E \text{ és } f(\vec{yx}) < c(\vec{yx})\}$$

és

$$B_{\text{hátra}} = \{x \in V \setminus S : \text{van olyan } y \in S, \text{ hogy } \vec{xy} \in E \text{ és } f(\vec{xy}) > 0\}.$$

Keressük meg  $B_{\text{előre}} \cup B_{\text{hátra}}$  egy  $x$  elemét.



# Keresés (folytatás)

## Algoritmus (folytatás)

Ezután három esetet különböztetünk meg:

- (i) **Bovítés:** Ha  $x \neq t$  akkor  $S \leftarrow S \cup \{x\}$  és folytassuk a Javítóút-kezdemények növelése lépéssel.
- (ii) **Sikeres keresés:** Ha  $x = t$  akkor „nyomozzuk vissza” hogyan jutottunk el ide. Az „ok” egy javító út lesz, leállunk a javító uttal.
- (iii) **Sikertelen keresés:**  $B_{\text{előre}} \cup B_{\text{hátra}} = \emptyset$ .  $t \notin S$  esetén juthatunk ide. Nem találtunk javító utat.

# A keresés visszakövetése

- Megjegyezzük, hogy a leírásban szerepel egy „visszanyomozás”.
- Érdekes  $S$ -ben egy fát tárolni. Ennek  $s$ -ből induló utai lesznek a megtalált javító út kezdemények.
- Minden bővítési lépés egy ághajtásos növeléssel megadja a nagyobb  $S$  nagyobb fáját. Az „ág” természetes a bővítő  $x$ -hez.
- Ha ebbe a fába bekerül  $t$ , akkor a fabeli  $st$  út lesz a megtalált javító út.

# Mi a sikertelen keresés?

Formálisan tudjuk, hogy kialakul egy  $S_{kifullad}$ , amelyre teljesül, hogy minden elhagyó élen az anyagmennyiség kapacitásnyi, továbbá minden belépő élen az anyagmennyiség 0.

# Vágás

## Definíció: Vágás

Legyen  $\vec{G}$  egy irányított gráf. Egy  $\mathcal{V} = \{S, T\}$  vágás  $\vec{G}$ -ben a ponthalmaz egy kétosztályú partíciója.  $S$  és  $T$  a vágás két osztálya vagy partja.  $\mathcal{V}$  egy **s-t vágás**, ha  $s \in S$ , és  $t \in T$ .

## Definíció: Vágás élhalmaza

$E(\mathcal{V})$  a  $\mathcal{V}$  vágás élhalmaza, azon élek halmaza, amelyek két végpontja a vágás két különböző oldalára esik.

Irányított gráfban  $E(\mathcal{V})$  természetes módon két osztályba sorolható aszerint, hogy a vágás egy élének kezdőpontja a forrás vagy a nyelő oldalán van.

$$E^+(\mathcal{V}) = \vec{E}(\mathcal{V}) = \{e = \overrightarrow{xy} : x \in S, y \in T\},$$

$$E^-(\mathcal{V}) = \overleftarrow{E}(\mathcal{V}) = \{e = \overleftarrow{xy} : x \in T, y \in S\}.$$

# Példa

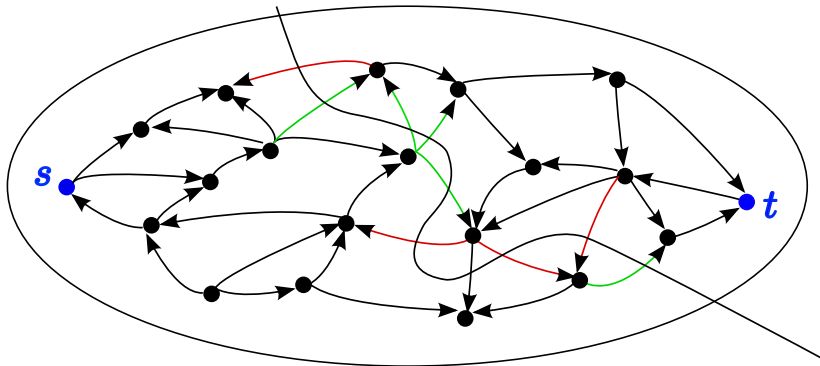


Figure: Egy vágás

# Folyamok értéke: Alternatívák

Lehetőség van a folyam értékének alternatív leírására és ez alapján a legnagyobb folyamértékére alternatív felső becsléseket kaphatunk.

## Lemma

Legyen  $f$  egy tetszőleges folyam.

$$(i) \quad \acute{e}(f) = \sum_{e:e \in E_{ki}(s)} f(e) - \sum_{e:e \in E_{be}(s)} f(e)$$

(ii) Tetszőleges  $\mathcal{V} = \{S, T\}$  s-t vágásra:

$$\acute{e}(f) = \sum_{e:e \in \vec{E}(\mathcal{V})} f(e) - \sum_{e:e \in \overleftarrow{E}(\mathcal{V})} f(e).$$

Tehát egy megfelelő  $\mathcal{V}$  vágás alapján ki lehet fejezni a folyam értékét: a forrás felől átfolyó anyagmennyiségből kivonva a nyelő irányából visszafolyó anyagmennyiséget megkapjuk  $\acute{e}(f)$ -t.

# Bizonyítás

(i) az (ii) pont speciális esete ( $S = \{s\}$ ,  $T = V(G) - \{s\}$ ), így elég (ii)-t igazolni.

$T$  minden  $v$  csúcsára egy egyenlőséget írunk fel. A  $v \in T \setminus \{t\}$  csúcsokra (azaz a nem forrásokra) felírjuk az anyagmegmaradás törvény rendezett formáját:

$$\sum_{e: e \in E_{be}(v)} f(e) - \sum_{e: e \in E_{ki}(v)} f(e) = 0.$$

A  $v = t$  esetben a folyam értékének definícióját írjuk fel:

$$\sum_{e: e \in E_{be}(v)} f(e) - \sum_{e: e \in E_{ki}(v)} f(e) = \epsilon(f).$$

Ezután összegezzük az összes  $T$ -beli csúcsra felírt egyenlőséget.

# Bizonyítás (folytatás)

A jobb oldal pontosan  $\epsilon(f)$  lesz.

Egy él viszonya a vágáshoz négyféle lehet.

Az  $S$ -beli élek nem szereplnek a felírt egyenlőségekben.

A  $T$ -beli  $e$  élek kettőben szereplnek. Az egyikben befutó élként, hozzájárulása az összeghez  $+$  előjellel/ $+1$  súllyal  $f(e)$ . Egy másikban kifutó élként, hozzájárulása az összeghez  $-$  előjellel/ $-1$  súllyal  $f(e)$ . Az összegzésben  $f(e)$  kiesik.

A maradék élek ( $\mathcal{V}$  élei) két kategóriába esnek:

- (1) A  $\vec{E}(\mathcal{V})$ -beli  $e$  élek egyetlen  $v$  csúcsra felírt egyenlőségben szereplnek és ott szerepük: befutó él. Az összeghez  $+f(e)$  hozzájárulást ad.
- (2) A  $\overleftarrow{E}(\mathcal{V})$ -beli  $e$  élek egyetlen  $v$  csúcsra felírt egyenlőségben szerepelnek és ott szerepük: kivezető él. Az összeghez  $-f(e)$  hozzájárulást ad.



# A Lemma következménye

Az alábbi becslést adhatjuk  $\epsilon(f)$ -re:

## Következmény

Legyen  $f$  tetszőleges folyam,  $\mathcal{V}$  vágás. Ekkor

$$\epsilon(f) \leq \sum_{e \in \vec{E}(\mathcal{V})} c(e) =: c(\mathcal{V}),$$

$c(\mathcal{V})$ -t a **vágás kapacitásának** nevezzük.

Az állítás bizonyítása egyszerű. A folyam érték  $\mathcal{V}$ -re alapuló felírásában az  $f(e)$  tagok  $c(e)$ -vel, a  $-f(e)$  tagok 0-val becsülhetők felül. A becslésünk tetszőleges folyamra és tetszőleges vágásra igaz.

# Egy második következmény

## Következmény

Ha Ford—Fulkerson javító út kereső algoritmus sikertelen kereséssel áll le, akkor nincs is  $f$ -re vonatkozó javító út a hálózatban.

Tudjuk, hogy sikertelen keresés végén kialakul egy  $\mathcal{V}_{\text{kifulladás}}$  vágás, amelyre teljesül, hogy minden előre vezető élén az anyagmennyiség kapacitásnyi, továbbá minden visszavezető élén az anyagmennyiség 0.

Ekkor azonban a fenti becslés egyenlőséggel érvényes, azaz  $é(f_{\text{aktuális}}) = c(\mathcal{V}_{\text{kifulladás}})$ .

De tetszőleges  $f$  folyamra  $é(f) \leq c(\mathcal{V}_{\text{kifulladás}})$ .

Azaz  $f_{\text{aktuális}}$  egy optimális folyam.

# Szünet



(2)

Az algoritmus javító utas növelésével próbálja elérni az optimális folyamot.

A fenti következmény csak azt mondta, ha az algoritmus leáll, akkor outputja korrekt.

Ciklizálhat-e az algoritmus? Azaz elképzelhető-e, hogy javítások végtelen sorozatát kapjuk, így sose érjük el az optimális folyamot.

Ciklizálás esetén a kapott folyam-sorozat értéke monoton nő és a hálózat által korlátozott. Azaz az értékek sorozata konvergens. A ciklizálásnak két kimenetele lehet.

Jobb esetben a kiszámolt folyamok értékei az optimális folyamértékhez konvergálnak. Elképzelhető-e, hogy az algoritmus ciklizál, de a folyamértékek sorozatának limesze nem a optimális folyam érték (hanem nyilván annál kisebb)?

# 1. válasz

- Valóságban a kapacitásfüggvény értékkészlete nem a pozitív valós számok halmaza, hanem  $\mathbb{Q}_{++}$ , azaz pozitív racionális számok halmaza.
- Ezek a kapacitás értékek (véges sok) skálázhatók úgy, hogy egészek legyenek (gondolhatunk arra, hogy alkalmas mértékegységváltást végzünk vagy a kapacitás értékeket leíró racionális számok közös nevezőjével mindent beszorzunk).
- Ha a kapacitások egészek és a kiinduló folyam is egészértékű (folyamot leíró függvény értékkészlete  $\mathbb{Z}_+ = \mathbb{N}$ ), akkor az algoritmus futása során végig csak egész számokkal dolgozik.
- Speciálisan  $\delta > 0$  is egész lesz, azaz  $\delta \geq 1$  (lásd a javítóút definícióját követő lemmát) Azaz a folyam minden javításánál a folyam értéke legalább 1-gyel nő, így nem lehet ciklizálás.

# title

## Tétel

Legyen  $\mathcal{H}$  egy hálózat, amelyre  $c : E(\vec{G}) \rightarrow \mathbb{Q}_{++}$  és  $f_0 : E(\vec{G}) \rightarrow \mathbb{Q}_+$ . Ekkor a Ford—Fulkerson-algoritmus véges lépésben megtalálja az optimális folyamot.

Azt is megjegyezzük, hogy a tétel nem kielégítő: Nem mond semmit arról hány javítás szükséges, azon túl hogy a javítások száma legfeljebb az optimális folyam értéke. Ez azonban hatalmas szám lehet. Ha a folyam kapacitásai  $k$  számjegyű számokkal adott, akkor az optimális folyam értéke lehet exponenciális  $k$ -ban.

Ennek ellenére a Ford—Fulkerson-algoritmus a legtöbb folyam algoritmus „keretét” adja.

# Egész értékűségi tétel

## Következmény

Legyen  $(\vec{G}, s, t, c)$  hálózat, amelyben  $c : E(\vec{G}) \rightarrow \mathbb{N}^+$ . Ekkor létezik  $f : E(\vec{G}) \rightarrow \mathbb{N}$  optimális folyam.

# Egész értékűségi tétel: Példa

Megjegyezzük, hogy nem állítjuk, sőt nem is igaz, hogy minden optimális folyam szükségszerűen olyan, hogy minden élen egész anyagmennyiség folyik. Az alábbi ábrán példák láthatók hálózatokra és egész, illetve nem egész optimális folyamokra.

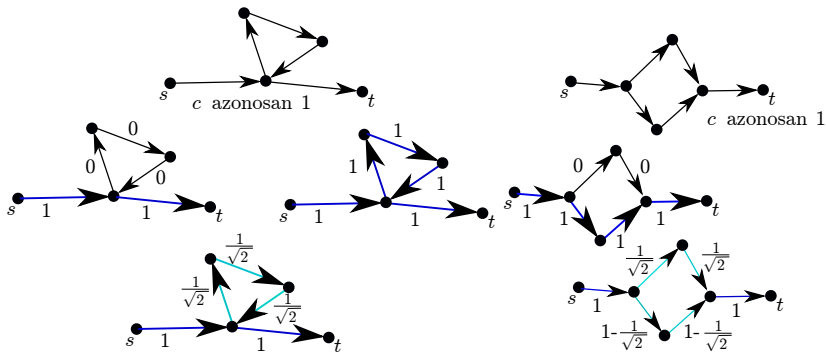


Figure:



## 2. válasz

- Elméletben elképzelhetjük, hogy pontos valós aritmetikával dolgozunk.
- A fenti algoritmus — javítóút-kezdemények növelése — olyan szabadon van megfogalmazva, hogy tetszőleges javítóút megtalálásához vezet.
- Azaz sok rövid javítóút létezése esetén is lehetséges, hogy a fenti nem-determinisztikus leírás egy hosszú javítóúthoz vezet.
- Példák adhatók, hogy ekkor elképzelhető az, hogy a kiszámolt folyamatok értékeinek monoton növekvő korlátos sorozata nem az optimális folyamértékhez tart.

# Folyamok alaptétele

## Folyamok alaptétele

Legyen  $f$  egy folyam a  $H(\vec{G}, s, t, c)$  hálózatban. A következő állítások ekvivalensek:

- (i)  $f$  folyam értéke maximális,
- (ii)  $f$ -hez van olyan  $\mathcal{V} = \{S, T\}$  vágás, hogy  $é(f) = c(\mathcal{V})$ ,
- (iii)  $f$ -hez nincs javító út.

# A bizonyítás

(i)  $\Rightarrow$  (iii): Tudjuk, javító út léte bizonyítja, hogy folyamunk nem optimális. A fenti implikáció ugyanezt az állítást tartalmazza átfogalmazva.

(ii)  $\Rightarrow$  (i): Tetszőleges  $\mathcal{V}$  vágás kapacitása tetszőleges folyam értékét felülről becsüli. Ha egy felső becslés egyben folyam érték is, akkor ez a folyam értéke biztos maximális.

(iii)  $\Rightarrow$  (ii) állítás. Ford—Fulkerson keresése kifulladás és definiál egy  $\mathcal{V}_{\text{kifulladás}}$  vágást, továbbá  $\epsilon(f) = c(\mathcal{V}_{\text{kifulladás}})$  teljesül. Ez bizonyítja  $f$  optimalitását.

# Emlékeztető

Tudjuk, hogy egy tetszőleges vágás kapacitása felülről becsli egy tetszőleges folyam értékét. Ennek az észrevételnek a „legerősebb” kihasználása:

$$\max_f \text{folyam } \dot{e}(f) = \min_{\mathcal{V}} \text{vágás } c(\mathcal{V}).$$

Azt is tudjuk, hogy optimális folyamra a Ford—Fulkerson keresés kifulladás és

$$\dot{e}(f_{\text{optimális}}) = c(\mathcal{V}_{\text{kifulladás}}).$$

# MFMC-tétel

## Maximális-folyam-minimális-vágás-tétel, MFMC-tétel

$$\max_f \text{folyam} \acute{e}(f) = \min_{\mathcal{V}} \text{vágás} c(V),$$

tehát a maximális folyam érték akkora mint a minimális vágáskapacitás.

A fentiek alapján érdemes a Ford—Fulkerson-algoritmust úgy módosítani, hogy leálláskor a kiszámolt  $\mathcal{V}_{\text{kifullad}}$  vágást is kiadja. Ez egy olyan vágás lesz, amely előremutató élein kapacitásnyi/maximális anyagmennyiség folyik, míg hátramutató élein nincs visszafolyás.

Ez egy laikus számára is mutatja az output korrektségét. Az outputban akkor is megbízhatunk (a megfelelő vágás láttán), ha az algoritmus kódolása esetleg nem megbízható.

# Vége van!

Köszönöm a figyelmet!