Algoritmuselmélet és bonyolultságelmélet MSc hallgatók számára

Javítgatásos algoritmusok, Ford—Fulkerson-algoritmus

2018. Előadó: Hajnal Péter

1. Javítgatásos algoritmusok

Egy egyszerű alapgondolattal kezdjük, majd egy példát mutatunk arra, hogyan lehet ezt egy konkrét algoritmikus problémánál használni.

Először az optimalizálás alapfelasatát vázolom. Legyen $\mathcal{L}$ lehetséges megoldások halmaza. Adott egy $c : \mathcal{L} \to \mathbb{R}$ célfüggvény. Szeretnénk meghatározni milyen nagy (bizonyos esetekben milyen kicsi) lehet $c(x)$, ahol $x \in \mathcal{L}$.

Egy természetes elindulás, hogy egy $x_0 \in \mathcal{L}$ értékkel kezdünk, majd próbálunk egy jobb $x_1 \in \mathcal{L}$ értéket találni, azaz „megjavítani" a korábbi értéket. Ha van egy szisztematikus módszerünk javításra, javító x keresésére, akkor megvan az algoritmusunk. Amíg lehet egy $x_0, x_1, \ldots, x_i, \ldots$ $\mathcal{L}$-beli sorozatot definiálunk. Mindegyik x_{i+1} az előző x_i javítása. reméljük, hogy valamelyik x_i a $\mathcal{L}$ halmaz egy optimális eleme lesz, vagy legalábbis $c(x_i)$ konvergál a lehető legnagyobb/legkisebb értékhez.

Példánk egy nagyon fontos probléma, a folyam-feladat lesz. Ehhez több fogalom bevezetése szükséges.

2. Hálózatok, folyamok

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

Megjegyzés. Hálózatok sok gyakorlati probléma absztrakciójához hasznosak. Például egy város vízvezetékhálózata 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.

A fenti fogalom egy statikus fogalom. Olyan mint egy programozó számára a hardwer. Az alkalmazott matematikus/programozó számára a hálózat adott/felmért és az alkalmazás inputja. A dinamika leírásához új fogalom kell.

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

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

(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. Ezek természetes fizikai feltételek.

Ahhoz, hogy könnyebben elképzeljük a folyamokkal kapcsolatos fogalmakat, nézzünk egy másik alkalmazást. A hálózat legyen egy város úthálózata. A forrás lehet egy lakótelepet, lakóparkot reprezentáló csúcs. A belvárost reprezentálja a nyelő. A folyam a reggeli forgalom: a belvárosba szeretnének eljutni autóval az emberek (azaz közbülső csúcsok nem „nyelnek el" forgalmat). Minden élen a folyam megadja az ott folyó forgalmat.

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 legyenes szükségünk van egy újabb fogalomra.

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

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

Folyam probléma: Adott egy hálózat. Keressünk hozzá egy maximális értékű folyamot.

A maximális jelző első olvasatban problémás. Egy folytonos problémával állunk szemben, amikor nem szüségszerű a legnagyobb érték fevétele. Egy folyam egy m élű hálózatban m valós szám leírásával adható meg, azaz azonosítható $\mathbb{R}^E \equiv \mathbb{R}^m$ egy pontjával. A folyamoknak 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.

Megjegyzés. Tulajdonképpen a folyam probléma a lineáris programozás feladat egy speciális esete: a maximalizálandó $\acute{e}(f)$ függvény lineáris, valamint a folyamok halmaza lineáris egyenlőségekkel és egyenlőtlenségekkel van definiálva.

Az, hogy az érték nem lehet tetszőleges nagy az elemi módon is könnyen látszik. Tetszőleges f folyam esetén

$$\acute{e}(f) = \sum_{e \in E_{be}(t)} f(e) - \sum_{e \in E_{ki}(t)} f(e) \leq \sum_{e \in E_{be}(t)} c(e).$$

Azaz $\acute{e}(f)$ a hálózat egy paraméterével becsülhető felülről.

⋆

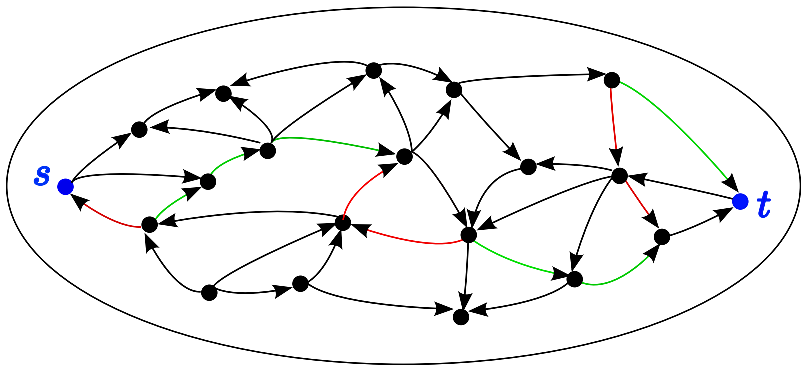
Definíció. Legyen $H(\overrightarrow{G}, s, t, c)$ hálózat, ebben pedig egy f folyam. Legyen P egy irányítatlan értelemben vett út $\overrightarrow{G}$-ben. (Azaz hagyjuk el a $\overrightarrow{G}$ gráf irányítását (így kapjuk a G irányítatlan gráfot. P egy út ebben.) P éleit két kategóriába sorolhatjuk ($\overrightarrow{G}$-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 $E(P) = E_{\text{előre}}(P) \bigcup E_{\text{hátra}}(P)$. Ekkor **P javítóút** ($f$ folyamra, $H(\overrightarrow{G}, s, t, c)$ 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 engedet maximumot (az él kapacitását) valamint visszafele (a hátramutató éleken) is történik bizonyos „visszafolyás”.

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



1. ábra. 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→ nyelő viszonylatban, míg a piros élek hátra mutatóak.

1. Lemma. *Legyen f egy folyam a $H(\overrightarrow{G}, s, t, c)$ 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**

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

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) $\text{é}(f^+) = \text{é}(f) + \delta$.

(2) és (3) együtt igazolja, hogy f^+ egy folyam. (1) és (4) együtt igazolja, hogy f^+ értéke nagyobb mint f-é. ■

Azaz egy javító út felismerése egy módot ad folyamunk javítására. Vajon ez a módszer univerzális-e? Azaz elképzelhető-e, hogy ne legyen javító út f-re de valami más módon mégis legyen nagyobb értékű folyam? Van-e ügyesebb módszer a folyamérték növelésére? Egy ilyen módszer javító út hiányában, egy más logika alapján adna nagyobb értékű folyamot.

Minden esetre a javítás fenti módszere ad egy lehetőséget egy algoritmus tervezésére.

3. Ford—Fulkerson-algoritmus

A folyamok alaptételéből adódik a következő séma:
`Ford--Fulkerson-algoritmus`

(I) **Inicializálás:** Vegyünk egy kiinduló f folyamot.
// Például a $f = 0$ folyamot (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”.

Több kérdés felmerül a fenti algoritmus leírásnál:

(1) Hogyan keressünk javító utat adott f-re. M'asképpen optimális folyam keresését lecseréltük javító út keresésre. Jó „üzlet” volt?

(2) Az algoritmus struktúrája megengedi a végtelen javítássorozatot. Lehet ciklizálás? Ha van ciklizálás akkor a kiszámolt folyamsorozat értékei egy monoton növő korlátos sorozatot adnak, azaz konvergál az érték. A limesz megegyezik a legnagyobb folyamértékkel?

(3) Mi a viszony a nem javítható folyamok és a maximális folyamok között?

4. A Ford–Fulkerson-algoritmus javító út keresése

A Keresés lépés az egyetlen algoritmikusan problémás lépés. Megoldása viszonylag egyszerű. Maga ez a része az algoritmusnak is javítgatásos. Egy $\overrightarrow{G}$-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 probáljuk hosszabbítani/javítani őket, remélve, hogy az egyik próbálkozás t-be vezet (és így javító utat találtunk).

Ford-Fulkerson javító út keresése:
`Keresés inicializálása`: Legyen $S := \{s\}$.
// S azon csúcsok halmaza, ahová javítóút-kezdeményeket találtunk.

`Javítóút-kezdemények növelése`:
// S növelése.
Legyen

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

és

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

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

Ezután három esetet különböztetünk meg:

(i) `Bővítés:` Ha $x \neq t$ akkor $S \leftarrow S \cup \{x\}$ és folyatssuk a `Javítóút-kezdemények növelése` lépéssel.

(ii) `Sikeres keresés:` Ha $x = t$ akkor „nyomozzuk vissza” hogyan jutottunk el ide. Az „ok” egy javító út lesz, léallunk a javító uttal.

(iii) `Sikertelen keresés:` $B_{\text{előre}} \cup B_{\text{hátra}} = \emptyset$. $t \notin S$ esetén juthatunk ide. Nem találtunk javító utat.

Megjegyezzük, hogy a leírásban szerepel egy „visszanyomozás”. Érdemes 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.

⋆

Vajon a sikertelen keresés mit jelent? Formálisan tudjuk, hogy kialakul egy $S_{kifullad}$, amelyre teljesül, hogy minden elhagyó élen az anyagmennyiség kapacitásnyi, továbbá minden belépő élen az anyagmennyiség 0.

További vizsgálatunk előtt bevezetjük a vágás fogalmát.

Definíció. Legyen $\overrightarrow{G}$ egy irányított gráf. Egy $\mathcal{V} = \{S, T\}$ vágás $\overrightarrow{G}$-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 $t \in T$.

Jelölés. $E(\mathcal{V})$ 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.

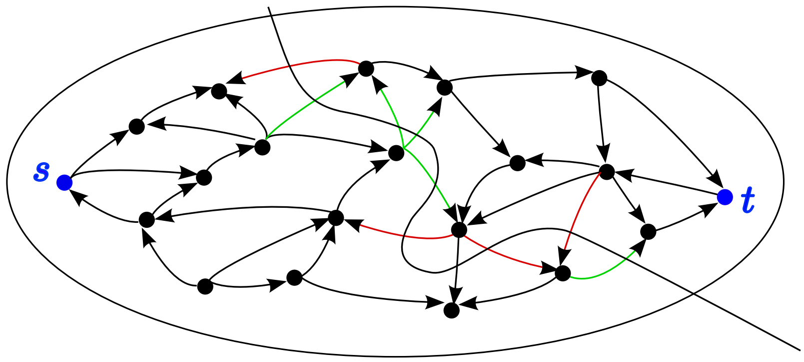
$$E^+(\mathcal{V}) = \overrightarrow{E}(\mathcal{V}) = \{e = \overrightarrow{xy} : x \in S, y \in T\},$$

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

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

2. Lemma. *Legyen f egy tetszőleges folyam.*

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



2. ábra. Egy vágás

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

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

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 é(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 $v \in T \backslash \{t\}$ 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:

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

Ezután összegezzük az összes T-beli csúcsra felírt egyenlőséget. A jobb oldal pontosan é(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 öszeghez + előjellel/+1 súllyal $f(e)$. Egy másikban kifutó élként, hozzájárulása az öszeghez − előjellel/−1 súllyal $f(e)$. Az összegzésben $f(e)$ kiesik. A maradék élek ($\mathcal{V}$ élei) két kategóriába esnek:

1. A $\overrightarrow{E}(\mathcal{V})$-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 egyenlóségben szereplnek és ott szerepük: kivezető él. Az összeghez $-f(e)$ hozzájárulást ad.

Így az összegzés után a bal oldalon a lemma állításában szereplő jobb oldali kifejezést kapjuk. Az állítás adódik. ■

Az alábbi becslést adhatjuk é(f)-re:

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

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

$c(\mathcal{V})$-t a **vágás kapacitás***ának 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.

4. Következmény. *Ha Ford—Fulkerson javító út kereső algopritmusa sikertelen kereséssel áll le, akkor nincs is f-re vonatkozó javíto út a hálózatban.*

Bizonyítás. Tudjuk, hogy sikertelen keresés végén kialakul egy $\mathcal{V}_{\text{kifullad}}$ 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$(f_{\text{aktuális}}) = c(\mathcal{V}_{\text{kifullad}})$. De tetszőleges f folyamra é$(f) \leq c(\mathcal{V}_{\text{kifullad}})$. Azaz $f_{\text{aktuális}}$ egy optimális folyam. ■

A fentiek után a szorgalmas hallgató kódolhatja az algoritmus kedvenc programozási nyelvében.

5. Ford—Fulkerson-algoritmus ciklizálása

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 folyamsorozat 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)?

A fenti kérdésekre többféle válasz is adható.

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. A következő tétel csak összegzi eddigi gondolatainkat.

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

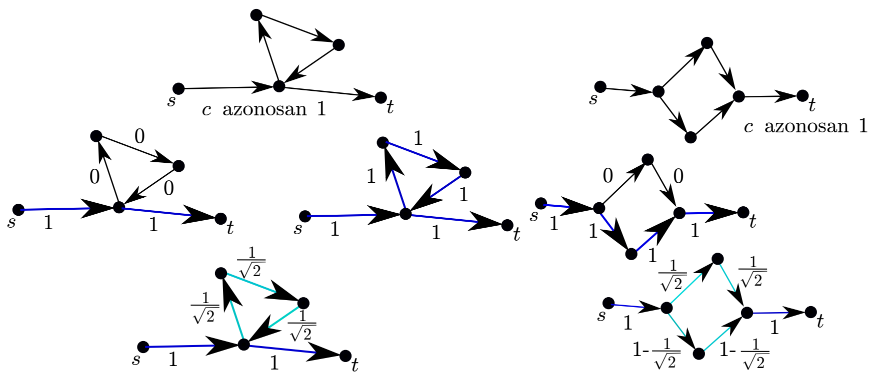
Azt is megjegyezzük, hogy a tétel nem kielégítő: Nem mond semmit arról hány javítás szükséges, azon túl hogy a javítások száma legfeljebb az optimális folyam értéke. Ez azonban hatalmas szám lehet. Ha a folyam kapacitásai k számjegyű számokkal adott, akkor az optimális folyam értéke lehet exponenciális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

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

A ciklizálás hiánya mellett egy másik eredmény is kijött. Érdemes ezt is kiemelni.

6. Következmény. *Legyen* $(\overrightarrow{G}, s, t, c)$ *hálózat, amelyben* $c : E(\overrightarrow{G}) \to \mathbb{N}^+$. *Ekkor létezik* $f : E(\overrightarrow{G}) \to \mathbb{N}$ *optimális folyam.*

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.



3. ábra.

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 folyamok értékeinek monoton növő korlátos sorozata nem az optimális folyamértékhez tart.

6. A folyamok alaptétele és következményei

Egy pillanatra megállunk és összgyűjtjük az eddig elért matematikai eredményeinket.

7. Tétel (Folyamok alaptétele). *Legyen* f *egy folyam a* $H(\overrightarrow{G}, s, t, c)$ *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.*

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

$(ii) \Rightarrow (i)$: 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—Fukerson keresése kifullad és definiál egy $\mathcal{V}_{\text{kifullad}}$ vágást, továbbá $\text{é}(f) = c(\mathcal{V}_{\text{kifullad}})$ teljesül. Ez bionyítja f optimalitását. ■

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

$$max_{f \text{ folyam}} \text{é}(f) = min_{\mathcal{V} \text{ vágás}} c(V).$$

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

$$\text{é}(f_{\text{optimális}}) = c(\mathcal{V}_{\text{kifullad}}).$$

Összegezve:

8. Tétel (Maximális-folyam-minimális-vágás-tétel, MFMC-tétel).

$$max_{f \ folyam} é(f) = min_{\mathcal{V} \ vágás} c(V),$$

tehát a maximális folyam érték akkora mint a minimális vágáskapacitás.

Megjegyzés. A fentiek alapján érdemes a Ford—Fulkerson-algorimtust 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, abban akkor is megbízhatunk, ha az algoritmus kódolása esetleg nem megbízható.